10-30-2020

A Taxonomy of Sequential Patterns Based Recommendation Systems

Hemni Sri Rajeswari Karlapalepu
University of Windsor

A Taxonomy of Sequential Patterns Based Recommendation Systems

By

Hemni Sri Rajeswari Karlapalepu

A Thesis
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science
at the University of Windsor

Windsor, Ontario, Canada

2020

A Taxonomy of Sequential Patterns Based Recommendation Systems

by

Hemni Sri Rajeswari Karlapalepu

APPROVED BY:

D. Yang

Department of Mathematics & Statistics

D. Alhadidi

School of Computer Science

C. Ezeife, Advisor

School of Computer Science

July 31, 2020

DECLARATION OF ORIGINALITY

I hereby certify that I am the sole author of this thesis. I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

With remarkable expansion of information through the internet, users prefer to receive the exact information they need through some suggestions to save their time and money. Thus, recommendation systems have become the heart of business strategies of E-commerce as they can increase sales and revenue as well as customer loyalty. Recommendation systems techniques provide suggestions for items/products to be purchased, rented or used by a user. The most common type of recommendation system technique is Collaborative Filtering (CF), which takes user's interest in an item (explicit rating) as input in a matrix known as the user-item rating matrix, and produces an output for unknown ratings of users for items from which top N recommended items for target users are defined. E-commerce recommendation systems usually deal with massive customer sequential databases such as historical purchase or click sequences. The time stamp of a click or purchase event is an important attribute of each dataset as the time interval between item purchases may be useful to learn the next items for purchase by users. Sequential Pattern Mining mines frequent or high utility sequential patterns from a sequential database. Recommendation systems accuracy will be improved if complex sequential patterns of user purchase behavior are learned by integrating sequential patterns of customer clicks and/or purchases into the user-item rating matrix input. Thus, integrating collaborative filtering (CF) and sequential pattern mining (SPM) of historical clicks and purchase data can improve recommendation accuracy, diversity and quality and this survey focuses on review of existing recommendation systems that are sequential pattern based exposing their methodologies, achievements, limitations, and potentials for solving more problems in this domain.

This thesis provides a comprehensive and comparative study of the existing Sequential Pattern-based E-commerce recommendation systems (SP-based E-commerce RS) such as ChoRec05, ChenRec09, HuangRec09, LiuRec09, ChoiRec12, Hybrid Model RecSys16, Product RecSys16, SainiRec17, HPCRec18 and HSPCRec19. Thesis shows that integrating sequential patterns mining (SPM) of historical purchase and/or click sequences into user-item matrix for collaborative filtering (CF) (i) Improved recommendation accuracy (ii) Reduced limiting user-item rating data Sparsity (iii) Increased Novelty Rate of the recommendations and (iv) Improved Scalability of the recommendation system. Thus, the importance of sequential patterns of customer behavior in improving the quality of recommendation systems for the application domain of E-commerce is accentuated through this survey by having a comparative performance analysis of the surveyed systems.

Keywords: sequential patterns, frequent patterns, sequential pattern mining, e-commerce, recommendations, recommender systems, collaborative filtering, clickstream history.

DEDICATION

I would like to dedicate this thesis to my parents, supervisor, internal and external readers and my friends who have helped and supported to complete my graduate study at the University of Windsor.

ACKNOWLEDGEMENTS

My sincere appreciation goes to my mentor Lord Sai Baba, parents Mr. Sesha Talpa Sayi and Mrs. Janaki. Your perseverance and words of encouragement gave me the extra energy to see this work through.

I will be an ingrate without recognising the invaluable tutoring and supervision from Dr. Christie Ezeife. Your constructive criticism and advice at all times gave me the needed drive to successfully complete this work. Thank you so much for your valuable time in reading all my thesis updates and providing me with the continuous financial support through Research Assistantship (R.A.) position.

Besides my advisor, I would like to thank my thesis committee members: my external reader, Dr. Dilian Yang, my internal reader, Dr. Dima Alhadidi and the chair, Dr. Saeed Samet for accepting to be my thesis committee, despite their tight schedules and their insightful comments and encouragement is highly appreciated.

Finally, I would express my appreciation to all my friends and colleagues at the University of Windsor, for their advice, encouragement and support throughout the duration of this work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

CHAPTER 1: INTRODUCTION

The increasing importance of the web as a medium for electronic and business transactions has served as a driving force for the development of recommendation systems which have become the heart of many Internet-based companies such as Google, YouTube, Facebook, Netflix, LinkedIn, Amazon, etc. Recommendation systems provide suggestions for items to be of use to a user. The suggestions provided are aimed at supporting their users in various decision-making processes, such as what items to buy, what music to listen, or what news to read (Ricci, Rokach, Shapira & Kantor, 2011). The entity (e.g., buyer/consumer) to which the recommendation is provided is referred to as the user, and the product (e.g., books) being recommended is referred to as an item. Recommendation systems use various sources of data (as input) to infer customer interests in order to generate meaningful recommendations to a user for items that might interest them (Felfernig, Friedric, Jannach, &, Zanker, 2011). Different types of recommendation systems take different input data and can belong in categories of 1) behavior pattern data 2) demographic data 3) production data 4) rating data and 5) transaction data as given in (Wei, Huang, & Fu, 2007) and with details of their form summarized in Table 1.1.

Table 1.1 Data used in recommendation systems (Wei, Huang, & Fu, 2007)

Data Type	**Description**
Behavior Pattern Data	Duration of browsing, click times, the links of webs; save, print, scroll, delete, pen, close, refresh of webs; selection, edition, search, copy, paste, bookmark and even download of web content and so on.
Demographic Data	Name, age, gender, profession, birthdate, telephone, address, hobbies, salary, education experience and so on.
Production Data	For movies or music, it means actor or singer, topic, release time, price, brand and so on, while for webs or documents, it means content description using key words, the links to others, the viewed times, the topic and so on.
Rating Data	Rating scores, such as discrete multi-levels ratings and continuous rating; and latent comments, such as best, good, bad, worse and so on.
Transaction Data	Purchasing date, purchase quantity, price, discounting and so on.

In order to acquire these inputs, there are two types of information gathering methods: explicit feedback (e.g., Table 1.4) and implicit feedback (e.g., Table 1.3). Explicit feedback includes collecting ratings of products or text comments by users through registration form/asking explicitly for interests and preferences in the form of ratings, where users select numeric values from a specific evaluation system (e.g., a five-star rating system) to specify their likes and dislikes of different items. However, implicit feedback is not quite as explicit but is easier to gather in the web-centered paradigm. This form of feedback includes behaviors such as purchase history, browsing history, search patterns, time spent on specific pages, links followed by a user, button clicks, user data from social network platforms. For example, the simple act of a user buying or browsing an item can be viewed as an endorsement of that item. Such forms of feedback are commonly used by online merchants such as Amazon.com (Aggarwal, 2016). Consider a user's click and purchase behavior data as shown in the Table 1.2 A sample user's click and purchase behavior data indicating that the customer ended up purchasing few items out of all the clicked items.

Table 1.2 A sample user's click and purchase behavior data

User Id	Click	Purchase
1	Cheese, Butter, Milk, Cream, Honey, Bread	Cream, Butter, Milk, Honey

Now, an implicit user's transaction (binary) matrix (Table 1.3) is created by analyzing the list of items purchased by the user and a value of 1 is assigned for the purchased items and 0 represents non purchased items by a user. Analyzing user's implicit preferences (i.e. the behavior pattern data) has been used widely and proved to be useful in practice in order to construct input user-item matrix when explicit rating information on items is not available.

Table 1.3 An implicit user-item matrix formed from Table 1.2

User/item	Milk	Bread	Butter	Cream	Cheese	Honey
User 1	1	0	1	1	0	1

The purpose of a recommendation system is often summarized as "help the users find relevant items", and the predominant operationalization of this goal has been to focus on the ability to numerically estimate the user's preferences for unseen items or to provide users with item lists ranked in accordance to the estimated preferences (Jannach & Adomavicius, 2016). The process of a recommendation system is that, it uses the known preferences of a group of users to make recommendations or predictions of the unknown preferences for other users (Su & Khoshgoftaar, 2009). Given an incomplete user-rating matrix R of m users for n items with missing ratings (r_{uj}) of item j for user u, the recommendation problem consists of predicting the rating value for user-item combination and is also referred to as the m x n matrix R completion problem for m users and n items

(Aggarwal, 2016). Let us consider a simple example of a movie recommendation to demonstrate this concept. For instance, in Table 1.4, each cell is the rating value (preference) of a user for a movie on a 5-point scale (i.e. from 1 to 5). The problem here is to predict a rating for a user on an un-rated item, such as preference level of David for the movie Fast & Furious, using a recommendation technique (e.g., CF algorithm in section 1.2.2). We obtain the output of the prediction as 4.55 (i.e. the predicted rating value for the unknown user-item combination David on Fast & Furious is 4.55) by computing the mean rating for each user, calculating the similarity between David and all other users using cosine similarity measure (Equation 1.2), computing the peer group of David based on the similarity values obtained and then predicting the unknown rating of David for movie Fast & Furious. Finally, with a predicted rating of 4.55, we can recommend Fast & Furious to David as the predicted rating value is higher (i.e. 4.55 out of 5 which is 91%) than the average rating value.

Table 1.4 An example User-item rating matrix for a movie recommendation site

User/Item	Terminator	Deadpool	Mission Impossible	James Bond	Fast & Furious
Alex	2	?	3	?	5
Bob	3	1	5	?	?
Catherine	1	?	?	3	4
David	2	4	1	1	? (4.55)

Data mining, also referred to as knowledge discovery from data (KDD), is a process of nontrivial extraction of implicit, previously unknown and potentially useful information (such as knowledge rules; constraints such as support, confidence; regularities or patterns such as frequent patterns, sequential patterns) from large amounts of data to guide decisions about future activities (Chen, Han, & Yu, 1996). The data sources can include databases, data warehouses, the web, other information repositories, or data that are streamed into the system dynamically. Common data mining tasks include pattern mining (which consists of discovering interesting, useful, and unexpected patterns in the databases), association rule mining, frequent pattern mining and sequential pattern mining (Han, Pei & Kamber, 2011) which are generally used by recommendation systems to generate a meaningful representation of user purchase data.

Sequential pattern mining (SPM) discovers interesting subsequences as patterns (Sequential patterns) in a sequence database that can be used later by end users or management to find associations between different items or events in their data for purposes such as marketing campaigns, business reorganization, prediction and planning in the domain of E-commerce.

A **Sequence database** stores a number of records, where all records are sequences $\{s_1, s_2, \ldots, s_n\}$ that are arranged with respect to time (Han, Pei & Kamber, 2011). A sequence database can be represented as a tuple <SID, sequence-item sets>, where SID: represents the sequence identifier and sequence-item sets specifies the sets of items enclosed in parenthesis ( ).

An example sequence database is retail customer transactions or purchase sequences in a grocery store showing, for each customer, the collection of store items they purchased every week for one month. For example, let us consider an example of historical daily purchase data of grocery store as shown in Table 1.5, which contains CustomerID to represent customers, PurchasedItem to represent a set of purchased items by customers and Timestamp to represent the time of purchase.

Table 1.5 Historical purchase data

CustomerID	**PurchasedItem**	**Timestamp**
01	Bread, Milk	10, Sep 2019 00:48:44
02	Bread	11, Sep 2019 10:48:44
01	Bread, Milk, Sugar	15, Sep 2019 10:48:44
02	Sugar, Tea	16, Sep 2019 09:48:44
01	Milk	18, Sep 2019 00:48:44
01	Tea, Sugar	19, Sep 2019 00:48:44

The sequential database can be constructed from historical purchase data by considering the period of time (day, week, and month). In this case, construct purchase sequential database from historical purchase data (Table 1.5) as presented in Table 1.6, where SID (01) contains < (Bread, Milk), (Bread, Milk, Sugar), (Milk), (Tea, Sugar)> which means customer (01) first purchased Bread and Milk together then purchased Bread, Milk and Sugar together in second purchase and Milk in third purchase, finally, Tea and Sugar together in the last purchase.

Table 1.6 Sequential database created from historical purchase data

SID	**Sequences**
01	< (Bread, Milk), (Bread, Milk, Sugar), (Milk), (Tea, Sugar)>
02	<(Bread), (Sugar, Tea)>

Sequential patterns are ordered set of items (events) that are occurring with respect to time (Agrawal & Srikant, 1996). A sequential pattern is denoted in the angular bracket (< >), and each itemset contains sets of items, where each item enclosed in parenthesis ( ) separated by commas represents a set of items purchased at the same time. For example, from Table 1.6 <(Bread), (Sugar, Tea)> is a sequential pattern which means, customer (02) first purchased Bread, finally Sugar and Tea together in the last purchase.

The problem of SPM can now be formally described as follows, Given:

(i) a set of sequential records (called sequences) representing a sequential database SDB $= \{s_1, s_2, \ldots, s_n\}$ with sequence identifiers 1, 2, 3, …. n

(ii) a minimum support threshold called min sup ξ and

(iii) a set of k unique items or events $I = \{i_1, i_2, . . ., i_k\}$;

SPM algorithms discover the set of all frequent sub-sequences S in the given sequence database SDB of items I at the given min sup ξ, that are interesting for the user. A sequence s is said to be a frequent sequence or a sequential pattern if it's support is greater than or equal to the minimum support (min sup ξ) (Mabroukeh & Ezeife, 2010).

With the increase in the use of world wide web for e-commerce businesses, there is a surge of interest in the design of recommendation systems that can potentially turn browsers into buyers by providing personalized recommendations to users by adapting to their taste and improving cross-sales and attaining customer loyalty (Schafer, Frankowski, Herlocker, & Sen, 2007). Therefore, recommendation systems have been integrated into all kinds of business since 1990s and has become the heart of business strategies of e-commerce out of all the other various recommendation domains (e.g., Movies, News etc.) discussed in the section below (section 1.1). The input of an e-commerce recommendation system, is usually a binary user-item rating matrix as the example in Table 1.7, only showing whether or not an item has been purchased or liked by a user previously. Thus, the user-item rating matrix can be extremely sparse and with low quality input data (less informative rating data which doesn't reflect much regarding: (1) how much a user likes an item; (2) how frequently or how long ago a user purchased an item; (3) what quantity of a product was purchased). One way to improve this input data is to integrate explicit rating with implicit rating drawn from historical purchase or click stream data or to use learning algorithms such as association rule (discussed in section 1.2.3.1), SPM, clustering (a process of grouping a set of related objects into subsets, where the objects in each subset share some similar patterns (observations, data items, or feature vectors) (Jain, Murty, & Flynn, 1999)) of historical data to extract clearer customer purchase and click data behavior to integrate into the user-item rating matrix to reduce the data sparsity and improve the recommendation quality and accuracy.

Table 1.7 An example User-item rating matrix for an E-commerce site

User Id/Products	Milk	Bread	Butter	Cream	Cheese
User1	1	1	1	?	1
User2	1	1	?	?	?
User3	1	?	?	1	1
User4	1	1	1	?	?

SPM can capture the customer purchase behavior over time using mined sequential patterns which is crucial since the time interval between items is useful to learn at what time next item might be purchased and the next purchase decision of a user is often influenced by their recent behaviors and considers the temporal preference of the user as a sequence of purchased items. An example frequent sequential pattern (FSP) that can be mined from a relevant E-Commerce purchase history sequential database is <(milk, bread), (milk, cream)> indicating that generally, it is learned from the historical purchase database that whenever customers buy milk and bread together in one week, they come back in the following week to buy milk and cream together. This sequential rule can be written as (milk, bread) → (milk, cream). With a sequential rule like this, some of the unknown ratings in the input user-item rating matrix of Table 1.7 can be filled such that all users who have purchased the antecedent items (milk, bread) have a higher chance of (say 0.5 or some more specific determined chance value) of purchasing also cream next. With this information, the ratings for users 1, 2 and 4 for cream can be changed from unknown to 0.5. In this way a sequential pattern can be used to improve on the quantity of rating values by providing the possible value for the missing/unrated item. A user-item purchase frequency matrix can then be constructed, where each value represents the quantity of a product purchased by a user. This purchase frequency is then normalized to a scaled value (0 to 1) representing how interested a user is in one item as compared to other items to improve rating quality. If these historical sequential purchase patterns of a user are analyzed and integrated into the user-item matrix input, the rating quality (specifying level of interest or value for already rated items) and quantity (finding possible rating for previously unknown ratings) can be enhanced and improved by using the mined sequential patterns (which will be discussed in detail in the next chapter 2). Thus, the recommendation quality can be improved in terms of accuracy, scalability and novelty. Therefore, this thesis focuses on sequential pattern-based recommendation systems with E-commerce as an application domain.

1.1 Various Recommendation System Application Domains

Recommendation systems have been developed in various domains such as Movies – Netflix (Salakhutdinov, Mnih & Hinton, 2007), News – Google (Liu, Dolan & Pedersen, 2010), Image – Tumblr (Shin, Cetintas, Lee & Dhillon, 2015), Video – YouTube (Covington, Adams & Sargin, 2016), Social Media – Facebook (Zuo, Zeng, Gong & Jiao, 2015), Travel – TripAdvisor (Lim, Chan, Leckie & Karunasekera, 2018), Music – Spotify (Chen, Moore, Turnbull & Joachims, 2012), E-commerce – Amazon (Jannach, Lerche & Jugovac, 2015). A summary of each of these various recommendation domains with respect to their input, output, common techniques used by these systems for the purpose of recommendation along with their examples is provided below (Table 1.8) followed by a brief insight about each of these domains.

Table 1.8 Summary of various recommendation domains in terms of their input, output, recommendation technique and example systems

Recommendation Domain	Input	Output	Recommendation Technique	Example System
Movie Recommendation	**Rating data** (5-star scale)	Recommends movie(s) to watch	Item-based Collaborative Filtering	**Netflix** (Salakhutdinov, Mnih & Hinton, 2007) **Movielens** (Wang, Shi & Yeung, 2015)
Music Recommendation	**Listening** & **Playlist data** which includes implicit contextual information about listening events (e.g., user, track, time, duration), explicit information about user preferences (e.g., loved tracks, playlists), and	Generates Playlists	Sequence modelling using Embedding methods	**Spotify** (Chen, Moore, Turnbull & Joachims, 2012) **SoundCloud** (Batmaz, Ali, Alper & Cihan, 2018)

	user listening sessions			
POI Recommendation	**Check-in history** (i.e., visits of users at different venues such as restaurants, hotels) **with the time stamp and location details**	Recommends Next User Location/ Recommends Next place to visit	Location-based Social Networks (LBSNs)	**Foursquare** (Cheng, Yang, Michael, Lyu & Irwin, 2013) **Gowalla** (Cheng, Yang, Michael, Lyu & Irwin, 2013)
Web Navigation Prediction	**Web log data** which includes information about the host IP address of the computer accessing the web page, the user identification number, the time of access, the unified reference locator (url) of the web page being accessed, the number of bytes of data being requested.	Finding user navigational patterns/ Next-page visit predictions	Web log pattern mining using Association Rules (AR), Sequential Patterns (SP), Contiguous and non-contiguous Sequential Patterns	**Amazon personalize** (Jindal & Sardana, 2020)
App Recommendation	**Spatiotemporal context data** from activity logs which includes last used app, user's current location, time and the user profile	Pre-fetch applications/ generate contextual suggestions on which app to use	Prediction models constructed using Bayesian methods	**Yahoo's Aviate** (Yates, Jiang, Silvestri & Harrison, 2015)
Book Recommendation	**Rating data**	Recommends Book(s) to read	Content based filtering using Latent Factor models	**Goodreads** (Shu, Shen, Liu, Yi & Zhang, 2017) **Amazon** (Shu, Shen, Liu, Yi & Zhang, 2017)

Social Networking Recommendation	**Tag information** such as user profiles, friends, followers, likes, comments and tags	Profiling users/ discovering hidden representation of users and reveal semantic relationships among items	Graph-based filtering	**Facebook** (Xu, Chen, Miao & Meng, 2017)
News & Article Recommendation	**Click logs** of users consisting of comments made by the user to a thread, timestamp for the interaction and history, tracking the number of clicks on articles classified into categories	Recommends topic categories such as politics, sports, finance etc of news/articles to watch/read	Content based filtering	**Google News** (Liu, Dolan & Pedersen, 2010)
E-commerce Recommendation	**Transaction data**/Purchase history; **Rating data**; **Clickstream data** containing sessions of clicks on items	Recommends next item(s)/product(s) to be purchased	Content based & Collaborative filtering techniques	**Amazon** (Jannach, Lerche & Jugovac, 2015)

1. Movie recommendation: The movie recommendation domain is the basis of recommendation systems research since there are many publicly available movies preference datasets of different volumes. Furthermore, the tabular structure of these datasets is well-suited for Collaborative Filtering tasks. The pioneer work in this field is for producing recommendations on Netflix dataset (Salakhutdinov, Mnih & Hinton, 2007). The movie recommendation dataset consists of ratings from randomly chosen, anonymous users on movie titles during a certain period. Based on the explicit movie ratings data, a user profile is generated, which is then used to make suggestions to the user. The system recommends an item based on the similarity between the content of the items being recommended and a user profile.

2. Music recommendation: With the recent advances, most people prefer to consume music digitally through online music services such as Spotify (Chen, Moore, Turnbull & Joachims, 2012) and SoundCloud (Batmaz, Ali, Alper & Cihan, 2018). The consumption of music is often session-based, and the listener's interest can change strongly from one session to another. The user experience can furthermore be influenced by the order in which the tracks are played, that is, weak ordering constraints can exist between the tracks. Such constraints can either be explicitly given by the user (Pauws, Verhaegh & Vossen, 2006) or can be inferred from listening logs of a user community as done in (Chen, Moore, Turnbull & Joachims, 2012). A music recommendation engine generates a playlist each week based on a user's listening habits by observing what bands and individual tracks the user has listened to on a regular basis.

3. Point-Of-Interest (POI) recommendations: Moving between places is a sequential process, where the user's movements are usually limited by distance, time or budget constraints. Recommendation systems have been applied in this context in different ways to predict the user's next location or make recommendations for the next place to visit, based on the user's current location (Cheng, Yang, Lyu & King, 2013). Considering several past user locations has shown to be helpful, for example, for travel planning or when predicting which place the user will most probably visit at a specific time (Lim, Chan, Karunasekera & Leckie, 2018).

4. Web navigation prediction: It is an early application area of recommendation systems (Zhou, Hui & Chang, 2004). Web browsing is usually a sequential process and next-page visit predictions provide users dynamic content tailored to an individual interest that fits their current browsing session. The personalization task generally takes the form of recommending one or more items/pages to a current user or to pre-load webpages possibly based on the patterns of past visitors who have similar profiles. These interesting usage patterns are derived from the data stored in web server or browser logs using either of the association rules, contiguous or non-contiguous sequential patterns.

5. App recommendation: App usage prediction is a more recent application field, where considering the user's current context is crucial. These systems use a prediction technique that exploits contextual information such as time, location and the user profile, to predict the user which app to use next. A typical example of a research work is described in (Yates, Jiang, Silvestri & Harrison, 2015), where repeated usage patterns are mined from activity logs to pre-fetch applications or to make contextual suggestions on which app to use.

6. Book recommendation: The book recommendation domain (Zhang, Yuan, Lian & Xie, 2016) is closely related to movie recommendation domain since both these end products have similar characteristics such as consuming period and content features. These systems need textual content since they rely on the descriptions of the items to provide users with recommendations. The latent factors from the user-item ratings are learnt to model the user preferences and to learn the item embeddings from the item description.

7. Social networking recommendation: A social networking platform allows users to stay in touch with their friends and meet new people with similar tastes. User profiles, friends, followers, likes, comments and tags constitute the terminology of recommendation in this domain (Xu, Chen, Miao & Meng, 2017). In these methods, the user social information contained in the complex networks plays a key role in obtaining the user's real demand to search for suitable products for the user. A weighted social interaction network is first mapped to represent the interactions among social users according to the gathered information about historical user behavior. Thereafter, the complete path set is mined by the graph mining algorithm to find social similar neighbors with tastes similar to those of the target user.

8. News and article recommendation: News and articles are usually large collections that are especially suitable for content-based recommendation. Besides news and articles (Ruocco, Skrede & Langseth, 2017), there are some other recommendable textual contents like blogs, tags, research papers, and citations. These recommendation systems build profiles of user's news interests based on their past click behavior and the user's current news interests are predicted from the activities of that user and the news trends demonstrated in the activity of all users based on the log analysis. The information filtering mechanism is combined with an existing content-based filtering mechanism using learned user profiles to generate personalized news recommendations.

9. E-commerce: The traditional form of commerce such as shopping in stores, consuming in restaurants, purchasing in malls has been cloned to internet as the mobile and computer technology developed. The term "e-commerce" refers to the activity of electronically buying or selling of products on online services or over the Internet. Recommendation Systems in e-commerce, helps to model the business process through analysis of customer requirements or their purchase behaviors (Schafer, Frankowski, Herlocker, & Sen, 2007). In the e-commerce application, recommendation systems can potentially turn browsers into buyers by providing personalized recommendations to users by adapting to their taste, thus improving cross-sales and attaining customer loyalty. From Amazon to online multimedia sites like Netflix and YouTube, recommendation systems have become an indispensable asset in many E-Commerce platforms to enhance their productivity.

In the past decades, due to rapid growth of internet usage, vast amount of data is generated and this growing nature of data with huge number of products (tens of thousands) being added on a daily basis makes the input user-item rating matrix data sparsity rate higher especially for the e-commerce domain as compared to other domains. For example, consider a commercial recommendation system such as book recommendation in Amazon.com. In these systems, even active customers may have purchased only under 1% of the products (1% of 2 million books is 20, 000 books) i.e. only a few of the total number of items available in a database are often rated by users (Sarwar, Karypis, Konstan & Riedl, 2000). Thus, in e-commerce recommendation systems, the number of ratings already obtained is usually very less when compared to the number of ratings that needs to be predicted. This results in a sparse user-item matrix and generates weak or poor recommendations as a result of insufficient rating information. Thus, E-commerce is the most often investigated domain in recommendation systems to improve the quality of recommendations and is also the focus of this survey.

Input data to an E-commerce recommendation system is either explicit ratings (e.g., Like/Dislike) or the ratings drawn implicitly from historical purchase data (e.g., purchase/non-purchase) which is not so informative and extremely sparse. Thus, the learning algorithms such as SPM should be used to mine the customer behavior (sequential patterns) from the historical purchase or click stream data in order to make the user-item rating matrix more informative and to make highly accurate recommendations. The output of an E-commerce recommendation system consists of predicting the unknown ratings of users for items from which top N recommended items for target users or top N recommended users for target items are derived.

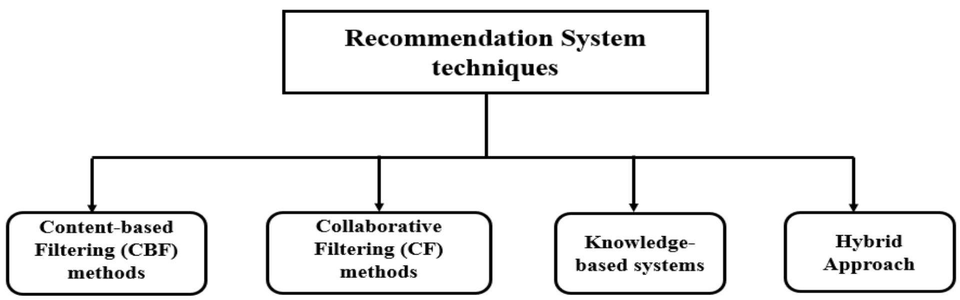
10. Other domains: Besides the domains described above, there exist substantial studies on **health care** (Zhao, Wang & Wang, 2015) – for disease diagnosis, most internet inquiry platforms provide similar cases recommendation where patients describe their illness and the doctors give professional suggestions online to avoid patient's longer waiting time, by finding the semantic similar cases which could be references in the large-scale historical case database; **accommodation** (Zhou, Albatal & Gurrin, 2016) - to recommend hotels based on the hotel features and the profiling of guest type (solo, family, couple etc) as additional information for personalized recommendation based on the previous information of the customer's reviews and ratings about the hotel's attributes like budget, location, room, food, cleanliness and facilities like pool, spa, gym, wifi etc.; **advertising** (Zhang et al., 2014) –

to recommend advertisements (ads) targeting to the search query based on user's behaviors in terms of the queries submitted, ads clicked or ignored, and the duration spent on the landing pages of clicked ads, etc. in the research area of recommendation systems.

1.2 Classification of Recommendation Systems techniques

With the growing importance of Recommendation Systems, fundamental knowledge and the early techniques for developing recommendation systems have been studied and depending on the techniques to be applied for recommendation systems, the types of information required for making recommendations are fairly different from one another. Based on how the recommendations are made, their goal and input data type, recommendation systems can generally be classified into four categories (Adomavicius & Tuzhilin, 2005): content-based filtering (CBF) methods, collaborative filtering (CF) methods, Knowledge-based systems and hybrid approach as shown in Figure 1.1.

Figure 1.1 Types of Recommendation System Techniques (Adomavicius & Tuzhilin, 2005)



While the CBF systems provide recommendations based on the user profiles (such as age, class) and product features (such as price, product attributes), CF systems does not consider these properties of items but uses only the preference (rating or voting) provided by users for items referred as rating matrix. With the knowledge-based systems, users explicitly specify their interests and the user specification is combined with domain knowledge to provide recommendations i.e. this approach relies on historical transaction data (i.e. purchase history) or/and click purchase data to use the knowledge about customers and the application domain for reasoning about what products fit the customer's preferences (Aggarwal, 2016). To avoid problems that exist in pure recommendation systems, hybrid solutions have been proposed which combines different recommendation system approaches. An overview of each of these recommendation techniques is provided in the following subsections.

1.2.1 Content-Based Filtering (CBF)

CBF technique is a domain-dependent approach that emphasizes on the analysis of attributes of the items in order to generate predictions. CBF recommendation systems (Chesnais, Mucklo & Sheena, 1995; Lang, 1995; Pazzani & Billsus, 2007) recommend items based on the similarity between items to recommend and items already purchased. In content-based methods, the ratings and buying behaviors of users are combined with the content information available in items. The term “content” refers to the descriptive attributes of items such as textual profiles or relevant keywords that are used for the purpose of recommendations. Thus, CBF recommendation systems, takes the rating matrix along with product specifications as input and predict the unknown ratings of user on item.

CBF systems typically:

(1) construct an item profile by extracting a set of features from each item in the item set
(2) build a content-based user profile from a set of features of the items that each user purchased
(3) calculate the similarity between the user profiles and the item profiles using a specific similarity function and
(4) recommend top n items with high similarity scores.

Problem 1.2.1: Predict the movie (Star Wars/Frozen) for recommendation, using Content-based filtering technique, given the reviews provided by a user for set of movies indicating the level of like (“Good”) or dislike (“Bad”).

Solution 1.2.1: The solution for Problem 1.2.1 is illustrated below:

Input: Item and User Profile data (Table 1.9) consists of a set of 4 movies and the column “Genre” correspond to features/attribute representing the category of a movie. The final columns of Table 1.9 contains the specified user taste, represented as “Good” or “Bad” and their corresponding ratings given on a rating scale of -10 to 10.

Output: User like or dislike (ratings) are not known for movies Star Wars and Frozen and thus needs to be predicted as to which movie should be recommended to the user.

Table 1.9 Item and User Profile Data

	Movies	**Genre (Attribute)**	**Rating**	**Review Given**
User	Mission Impossible	Action	9	Good
	Toy Story	Kids	-6	Bad
	Star Wars	Action	Recommend (Yes/No)?	
	Frozen	Kids	Recommend (Yes/No)?	

Step 1 - Item Profile Construction: The column "Genre" corresponds to feature/attribute representing the category of a movie. Construct an item vector in the order of movie genres (Action & Kids) which would be (1,0) for movies Mission Impossible & Star Wars (as the movie falls under Action genre but not Kids genre, hence value 1 for Action & 0 for Kids) and similarly (0,1) for movies Toy Story & Frozen.

Step 2 - User Profile Construction: All the rows of Table 1.9 correspond to movies and the user preference indicates user loves Action movies (as the user has given a good review and a rating value of 9 out of 10 for action movies) over Kids movies. Construct a user vector in the order of movie genres (Action & Kids) which would be (9, -6) which are the ratings specified by the user for the corresponding movie genres.

Step 3 – Computation & Recommendation: We now consider the dot product of two 2-D vectors, which are the User and Item Vectors in order to find the similarity.

Dot product of 2-d vectors $v_1 = (x_1, y_1)$ and $v_2 = (x_2, y_2)$ is $v_1.v_2 = x_1x_2 + y_1y_2$, where v_1 is the user vector and v_2 is the item vector.

For the movie "Frozen", $v_1 = (9, -6)$ and $v_2 = (0, 1)$. So, the dot product is $9 * 0 - 6 * 1 = -6$.

Similarly, for the movie "Star Wars", $v_1 = (9, -6)$ and $v_2 = (1, 0)$. So, the dot product is $9 * 1 - 6 * 0 = 9$.

As the rating value obtained for "Star Wars" is higher than the value obtained for the movie "Frozen", Star Wars will be recommended to the user, which also matches the intuition that user likes Action movies when compared to the kid's genre. In a similar manner, we can calculate the dot product of all the item vectors of all the movies in-store and recommend top 10 movies to the user.

CBF systems, however, have several **limitations**:

(1) it is not easy to obtain enough features to build profiles (insufficient features problem)

(2) recommended items are limited to those that are similar to the items that a target user purchased in the past (over-specialization problem) and

(3) new users who have not purchased items or users unusual in their preference cannot get a proper recommendation (new or unusual user problem)

1.2.2 Collaborative Filtering (CF)

CF technique uses the known preferences of a group of users to make recommendations or predictions of the unknown preferences for other users (Su, & Khoshgoftaar, 2009). Thus, the term "collaborative filtering" refers to the use of ratings from multiple users in a collaborative way to predict the unknown ratings.

CF-based recommendation systems typically:

(1) build a user profile from rating information of each user on items.

(2) identify like-minded users who rate items similar to a target user, using a similarity function such as cosine similarity, Pearson correlation coefficient or distance-based similarity and

(3) recommend top n items that the like-minded users preferred after their ratings are predicted as an average, weighted sum or adjusted weighted sum of ratings given on items by the identified like-minded users.

These methods of rating prediction are called memory-based/neighborhood based collaborative filtering algorithms because the ratings of user-item combinations are predicted based on their neighborhoods (Joaquin & Naohiro, 1999; Si & Jin, 2003). Memory-based CF can be achieved in two ways: through user-based and item-based techniques.

User-based collaborative filtering: Similarity functions are computed between the rows of the ratings matrix to discover similar users. In other words, the ratings provided by like-minded users of a target user 'A' are used in order to make the recommendations for 'A'. Thus, the basic idea is to determine users, who are similar to the target user 'A' and recommend ratings for the unobserved ratings of 'A' by computing weighted averages of the ratings of this peer group.

Item-based collaborative filtering: Similarity functions are computed between the columns of the ratings matrix to discover similar items, i.e. to determine the rating predictions for target item 'B' by user 'A', first find a set S of items that are most similar to target item 'B'. The ratings in item set S, which are specified by 'A', are used to predict whether the user 'A' will like item 'B'.

Problem 1.2.2: Predict the unknown rating value i.e. the rating of User C on Items 1 & 6 using Collaborative filtering technique, given the reviews (explicit ratings) provided by the users for set of items on a 7-point scale indicating the level of like or dislike for the item (Aggarwal, 2016).

Solution 1.2.2: The solution for Problem 1.2.2 using a user-based CF model is illustrated below:

Input: A user-item rating matrix (Table 1.10) where the ratings specified are on a 7-point scale {1, 2, 3, 4, 5, 6, 7} indicating the specific level of like or dislike of item representing from left to right, extreme dislike to extreme like.

Output: To predict a rating of User C on Item 1 and Item 6 using collaborative filtering.

Table 1.10 User-item rating matrix (Aggarwal, 2016)

	Item 1	**Item 2**	**Item 3**	**Item 4**	**Item 5**	**Item 6**	**Mean rating**
User A	7	6	7	4	5	4	33/6
User B	6	7	?	4	3	4	24/5
User C	?	3	3	1	1	?	8/4
User D	1	2	2	3	3	4	15/6
User E	1	?	1	2	3	3	10/5

Step 1: Compute the mean rating for User A, User B, User C, User D, and User E using all their rated items.

Equation 1.1: Equation to compute mean rating

Mean rating (r_u) = $\sum_{i \in I} r_{ui}$ / $|Number\ of\ items|$

For User A = (7+6+7+4+5+4)/6 = 33/6 = 5.5. Similarly, User B = 24/5 = 4.8, User C = 8/4 = 2, User D = 15/6 = 2.5 and User E = 10/5 = 2.

Step 2: Compute similarity between User C and other users. The similarity between User C and all other users can be computed using Cosine similarity or Pearson-Correlation Coefficient. In our case, we have used Cosine similarity, which is calculated using the below equation.

Equation 1.2: Formula to Compute Cosine similarity

$$Cosine\ (u, v) = \frac{\vec{u}.\vec{V}}{\|u\|.\|v\|} = \frac{r_{u1}.r_{v1} + r_{u2}.r_{v2} + \cdots + r_{un}.r_{vn}}{\sqrt{r_{u1}^2 + r_{u2}^2 + \cdots + r_{un}^2} * \sqrt{r_{v1}^2 + r_{v2}^2 + \cdots + r_{vn}^2}}$$

For example, SIM (User A, User C) = (6∗3+7∗3+4∗1+5∗1) / ($\sqrt{}6^2+7^2+4^2+5^2$) ∗ ($\sqrt{}3^2+3^2+1^2+1^2$) = 0.956. Similarly, SIM (User B, User C) = 0.981, SIM (User D, User C) = 0.789 and SIM (User E, User C) = 0.645.

Step 3: Select Top-N (in our case N=2) neighbors of User C by comparing their Cosine similarity. In our case, User A and User B have the highest similarity with User C. So, they are selected as Top-2 neighbors.

Step 4: Compute the raw rating value using Top-N users (User A and User B). To compute raw rating, Top-N users rating on item are used. For example, Raw rating $_{\text{User-C, Item1}}$ is calculated by using rating of User A on Item 1 and rating of User B on Item 1.

Raw rating $_{\text{User-C, Item 1}}$ = (7 ∗ 0.956 + 6 ∗ 0.981) / (0.956 + 0.981) = 6.49

Raw rating $_{\text{User-C, Item 6}}$ = (4 ∗ 0.956 + 4∗ 0.981) / (0.956 + 0.981) = 4

Step 5: Compute mean centric rating.

From the above raw ratings obtained in step 4, we can see that Item 1(rating value of 6.49) should be prioritized over item 6 (rating value of 4) to recommend to User C. Furthermore, the prediction suggests that User C is likely to be interested in both Item 1 and Item 6 to a greater degree than other items. Thus, mean centric rating needs to be computed to remove the bias. The mean centric rating helps to reduce the influence caused by high and low rating provided by users on items. For example, mean centric rating of User A on Item 1 is computed by subtracting rating of User A on Item 1 and mean rating of User A (in our case, 7-5.5=1.5).

Mean centric rating $_{\text{User-C, Item 1}}$ = $2 + \frac{1.5*0.956+1.5*0.981}{0.956+0.981} = 3.35$

Mean centric rating $_{\text{User-C, Item 6}}$ = 0.86.

CF recommendation systems, however, also have some **limitations**:

(1) it is difficult to recommend items for users who have never rated items before (new user problem)

(2) it is difficult to recommend items which have never been rated before (new item problem) and

(3) they make poor recommendations when rating information is insufficient (sparsity problem).

(4) As the number of users and products grow rapidly, the time complexity and space complexity issues become more prominent (scalability issue)

The New user and/or New item problem is technically referred to as Cold-start problem; a situation where a recommendation system does not have adequate information about a user or an item in order to make relevant predictions.

1.2.3 Knowledge-based Systems

Knowledge-based recommendation systems (Burke, 2002; Bridge, Goker, McGinty & Smyth, 2005; Felfernig & Burke, 2008) are particularly useful in the context of items that are not purchased very often. Also, an item may have attributes associated with it that correspond to various properties, and a user may be interested only in items with specific properties. Thus, in these cases, the item domain tends to be complex in terms of its varied properties, and it is hard to associate sufficient ratings with the large number of combinations at hand. Such cases can be addressed with knowledge-based recommendation systems, in which the recommendation process is performed based on similarities between customer requirements and item descriptions, or the use of constraints specifying user requirements. The process is facilitated with the use of knowledge bases (the approach takes its name from this fact) which contain data about rules and similarity functions to use during the retrieval process.

Knowledge-based recommendation systems can be further classified on the basis of the type of corresponding knowledge used to achieve the recommendation process:

1. Case-based recommendation systems: In case-based recommendation systems (Burke, 2002; Bridge, Goker, McGinty & Smyth, 2005), specific cases (examples) are specified by the user as targets or anchor points. Similarity metrics are defined on the item attributes to form the domain knowledge to retrieve similar items to these cases.

2. Constraint-based recommendation systems: In constraint-based recommendation systems (Felfernig & Burke, 2008; Felfernig, Friedrich, Jannach & Zanker, 2011), users typically specify requirements or constraints (e.g., lower or upper limits such as support threshold) on the item attributes. Domain-specific rules (e.g., association rules or sequential rules/patterns) that could take the form of domain-specific constraints on the item attributes are used to match the user requirements to item attributes (e.g., of a rule: Given a historical purchase data with a list of 5 items (a, b, c, d & e) and 4 transactions, how often (support) are the items a & d purchased together). Depending on the number and type of results returned, the user will have an opportunity to modify their original requirements (either relax some of the constraints when too few results are returned or add more constraints until the user arrives at desired results).

1.2.3.1 Association rule mining

Association rule mining is a data mining technique that falls under the category of Constraint-based knowledge recommendation system. The objective of Association rule mining is to find the co-occurrence relationships called associations, among the attribute values of tuples in a customer transaction database (Liu, Liao & Choudhary, 2005). A transaction database is a set of records (transactions) indicating the items purchased by customers at different times. The classic application of association rule mining is the market basket analysis using the frequent pattern mining (which is to discover frequent itemsets, a group of values/items that have occurred at least as frequently in the database as the given minimum support) algorithm such as Apriori (Agrawal & Srikant, 1994), which aims to discover how items purchased by customers in a supermarket are associated.

Consider a transaction database T = $\{T_1, T_2, \ldots T_m\}$, containing m transactions, which are defined on n items I. Therefore, I is the universal set of items, and each transaction T_i is a subset of the items in I. The key in association rule mining is to determine sets of items that are closely correlated in the transaction database. This is achieved with the notions of support and confidence, which are the measures that quantify the relationships between sets of items.

Support: The support of an itemset $X \subseteq I$ is the fraction of transactions in T, of which X is a subset. The support count of Itemset (set of items purchased in each transaction) in transaction database is the number of transactions in the database that contain the itemset. It can be defined as the number of tuples or the percentage of the database tuples in the table that contains these set of items.

Equation 1.3: Equation to compute support of an itemset

$$Support\ (itemset) = \frac{number\ of\ tuples\ in\ the\ itemset}{total\ number\ of\ tuples\ in\ the\ database}$$

To illustrate the definition, consider an example customer transaction database depicted in Table 1.11. For example, if we're interested in finding the support of itemset {bread, butter} from Table 1.11,

$$Support\ (Bread\ \&\ Butter) = \frac{3}{4} = 75\%.$$

If the support of an itemset is at least equal to a predefined threshold ξ, then the itemset is said to be frequent and is referred to as frequent itemset or frequent pattern and this threshold is referred to as the minimum support. These frequent itemsets can provide important insights about correlations in customer buying behavior and such inferences are very useful from the point of view of a recommendation system.

Table 1.11 Customer Transaction Database

Transaction ID	**Set of items purchased**
T_1	Bread, Butter, Milk
T_2	Bread, Butter
T_3	Bread, Butter, Milk, Sugar
T_4	Milk, Sugar

Confidence: An association rule is denoted in the form X ⇒ Y, where "⇒" is intended to give a direction to the nature of correlation between the set of items X and Y. The confidence of the rule X ⇒ Y is the conditional probability that a transaction in T contains Y, given that it also contains X. Therefore, the confidence is obtained by dividing the support of X & Y with the support of X. The strength of a rule is measured by its confidence. The Confidence of a rule is defined as the percentage of transactions in a database that contain the set of items in the right-hand side of the rule along with the items on the left-hand side.

For example, from Table 1.11,

Equation 1.4: Equation to compute confidence of a rule

$$Confidence\ (Bread\ \Rightarrow\ Butter)\ =\ \frac{|\,Bread\ \&\ Butter|}{|\,Bread|}\ =\frac{3}{3}\ =\ 100\%.$$

1.2.3.2 Sequential Pattern Mining (SPM)

Sequential Pattern mining is a data mining technique that falls into the class of constraint-based Knowledge systems under the category of Knowledge-based recommendation systems in our classification. Sequential pattern mining (SPM) discovers frequent subsequences as patterns (sequential patterns) in a sequence database. A sequence database stores a number of records, where all records are sequences of ordered events, with or without concrete notions of time. An example sequence database is retail customer transactions or purchase sequences in a grocery store showing, for each customer, the collection of store items they purchased every week for one month. These sequences of customer purchases can be represented as records with a schema [Transaction/Customer ID, <Ordered Sequence Events>], where each sequence event is a set of store items like bread, sugar, tea, milk, and so on. An example purchase sequential database with one such customer is [T_1, <(bread, milk), (bread, milk, sugar), (milk), (tea, sugar)>]. This sequential purchase pattern can be interpreted as, the customer made a purchase each of the four weeks in a month and first purchased Bread and Milk together then purchased Bread, Milk and Sugar together in second purchase and Milk in third purchase, finally, Tea and Sugar together in the last purchase. Other examples of sequences are DNA sequences and web log data. SPM is an important problem with broad applications, including the analysis of customer purchase behavior, web access patterns, scientific experiments, disease treatment, natural disasters, and protein formations. An SPM algorithm mines frequent sequential patterns from a sequential database as sequences with support greater than or equal to a given minimum support that can be used later by end users or management to find associations between the different items or events in their data for purposes such as marketing campaigns, business reorganization, prediction and planning.

SPM algorithms can be used for recommendations on their own, such as in the area of Web Recommendation Systems (WRS) which rely on the history and behavior of users to recommend future item purchases and page views. WRS are built on top of Web Usage Mining (also called web log mining) which is an important application of sequential pattern mining concerned with finding user navigational patterns on the world wide web by extracting knowledge from web logs. Similarly, SPM algorithms can be used in several other domains (section 1.1) for the purpose of recommendations.

Knowledge-based systems are unique in that they allow the users to explicitly specify what they want, and this explicit specification of requirements results in greater control of users over the recommendation process. Unlike CF recommendation systems which are solely dependent only on explicit rating data for the purpose of recommendations, Knowledge-based systems are based on historical transaction data and click purchase data which results in better recommendations as it captures better customer behavior. The **limitation** of Knowledge-based systems is that the constructed model is specific to the user at hand, as the use of community (peer) ratings is not leveraged (Aggarwal, 2016). This phenomenon tends to reduce the diversity of the recommended items, which is undesirable (over-specialization problem).

1.2.4 Hybrid Methods

To avoid problems that exist in CBF, CF and Knowledge-based systems, hybrid filtering technique has been proposed. Hybrid systems (Balabanović & Shoham, 1997; Salter & Antonopoulos, 2006) combine the strengths of different recommendation system techniques discussed so far, to create a technique that can perform more robustly and gain better system optimization to avoid some limitations and problems of the individual recommendation system techniques. The idea behind hybrid techniques is that a combination of algorithms will provide more accurate and effective recommendations than a single algorithm as it combines the strengths of various types of recommendation systems and the disadvantages of one algorithm can be overcome by another algorithm (Schafer, Frankowski, Herlocker & Sen, 2007), thus performing more robustly in a wide variety of settings.

The combination of approaches into a hybrid recommendation system can be done in either of the following ways:

(1) Implementing the approaches (CF and SPM (knowledge-based)) separately and combining their predictions e.g., LiuRec09 & ChoiRec12 systems
(2) Constructing a general unifying model that incorporates the characteristics of both the approaches (both CF and SPM (knowledge-based) characteristics e.g., ChoRec05, HuangRec09, Hybrid Model RecSys16, Product RecSys16, HPCRec18 and HSPRec19 systems)

The **limitation** of hybrid recommendation systems includes the problem of finding the best way of combining the predictions of different recommendation techniques employed and of determining appropriate weights for each individual method in the final prediction when designing a hybrid recommendation system.

An important task for e-commerce sites is to make predictions about what users might buy in future, based on the user's history of shopping. This problem can be modeled by using either of the recommendation techniques discussed so far. One of the most successful method in the literature is the CF technique which makes use of explicit data from user for the purpose of recommendation. A major advantage of this model is its ability to capture general taste for recommendation. However, this kind of algorithm has two obvious shortcomings. First, the effectiveness of such algorithms will be greatly reduced when the user's explicit behavior data is sparse, the second is these methods ignore the time context of user behavior (how the customer’s purchase behavior may vary over time), i.e. they are unable to capture the sequential behavior of users. SPM technique of Knowledge-based systems, therefore, has been used recently to make the recommendations more effective by extracting sequential patterns of user purchase behavior because the user's next purchase will be affected by its previous actions. This recommendation often utilizes user's implicit feedback data and the major advantage of this model is its ability to capture sequential behavior for recommendations. However, this model cannot capture a user's general taste. It can be seen that both of the methods have some defects. In fact, both sequential behavior and user's general taste are important factors that influence user's purchasing behavior. This motivates us to conduct a systematic review on the importance of integrating SPM with CF for recommendation systems, to improve the recommendation quality through more diverse recommendations, closing the high sparsity matrix problem and thus, making recommendations better by taking into account the user's general taste and sequential behavior.

1.3 Need for Sequential Pattern Mining in E-commerce Recommendation

1) User-Item interactions are sequentially dependent

In E-commerce recommendation systems, the crucial task is to identify the next item that a user will purchase. This next purchase decision of a user is often influenced by their recent behaviors (Li, Niu, Luo, Chen & Quan, 2019). For example, after buying a SLR camera, the user would be highly interested next in purchasing camera lenses. The U-I (user-item) ratings matrix encodes the individual preferences of users for items in a collection and provides the basis framework for CBF & CF techniques. Though these traditional methods can effectively model user preferences, they have a major drawback: They fail to account for sequential dynamics, providing a static list of recommendations regardless of a user's sequence of recent interactions, which results in missing the user's preference shift through the time.

For example, consider the events of a user called Jimmy,

Jimmy → Flight Booking → Hotel Booking → Rent a car → ?

Before Jimmy started holiday, he booked a flight and a hotel and rented a car successively. Now, what will be his next action? His next action may be visiting a tourist attraction via self-driving. In such a case, the hotel may be close to the destination airport of the flight and the location for picking up the rented car may be not far away from the hotel. In this scenario, each of Jimmy's next actions depends on the prior ones and thus all the four consumption actions are sequentially dependent. Such kind of sequential dependencies commonly exist in transaction data but cannot be well captured by the conventional content-based or collaborative filtering recommendation systems, as these recommendation techniques doesn't integrate historical purchase data which captures the customer behavior well.

This essentially led to the development of Sequential pattern-based recommendation systems. These systems suggest items that may be of interest to a user by mainly modelling the sequential dependencies over the user-item interactions in a sequence (Wang, Cao & Wang, 2019).

2) Improve the quality and quantity of ratings

Recommendation systems in E-commerce suffer from uninformative rating data which usually only represents if a user has purchased a product before, this user-item rating matrix is usually sparse, less informative and leads to poor recommendations (Bucklin & Sismeiro, 2003). Thus, in order to capture more real-life customer purchase behavior and to provide the relationship between already purchased items and recommended items, historical sequential purchase patterns of a user are analyzed and integrated into the user-item matrix input to enhance and improve the rating quality and quantity by providing the possible value for the missing/unrated item. To demonstrate this, consider a historical purchase data (Table 1.12)

Table 1.12 Historical purchase data

User Id	Purchase items	Timestamp
User1	Cream, Butter, Milk	2017.06.05.13.38.00
User1	Honey, Butter	2017.06.06.09.40.20
User2	Butter, Cheese	2017.06.05.19.40.16
User 2	Cheese, Honey	2017.06.06.10.40.16

Step 1: Create a user-item purchase frequency matrix (Table 1.13) from the historical purchase data (Table 1.12), where the values indicates the number of times an item was purchased by a user. For example, User 1 purchased butter twice, Honey once and so on.

Table 1.13 User-item frequency matrix created from historical purchase data

User/item	Milk	Bread	Butter	Cream	Cheese	Honey
User 1	1	?	2	1	?	1
User 2	?	?	1	?	2	1

Step 2: Now, convert historical purchase data (Table 1.12) to a sequential database (Table 1.14) by considering the period of time (day, week, and month) of the purchase.

Table 1.14 Purchase sequential database created from historical purchase data

SID	Purchase sequence
1	< (Cream, Butter, Milk), (Honey, Butter)>
2	< (Butter, Cheese), (Cheese, Honey)>

Step 3: Create frequent sequential purchase patterns from the sequential database (Table 1.14) using any SPM algorithm like GSP and the possible purchase sequential rules (Table 1.15) from frequent purchase sequences are extracted. Using these sequential purchase rules, some of the unknown ratings in user-item purchase frequency matrix (e.g. value of User1 for item cheese in Table 1.13) can be filled by using a predicted value such that all users who have purchased the antecedent items like (milk, butter) from Rule No:1 of Table 1.15 have a higher chance of (say 0.5 or some more specific determined chance value for the highly probable purchases determined by the SPs) purchasing also cheese next. Hence, using Rule No:1 it can be inferred that as the user1 purchased milk and butter in

this transaction, there are high chances that he would even purchase cheese in the same transaction. Hence, assign a value of 0.5 to the user-item combination (User1-Cheese). Similarly, (User2-Cream) is filled using Rule No:3 and (User2-Milk) is filled using Rule No:2.

Table 1.15 Sequential rules created from n-frequent sequences

Rule No	Sequential rule
1	Milk, Butter → Cheese
2	Cream, Cheese → Milk
3	Cheese, Honey → Cream

Step 4: The final enriched user-item frequency matrix created with help of sequential rules as described above is shown in Table 1.16.

Table 1.16 Rich user-item frequency matrix created with help of sequential rule

User/item	Milk	Bread	Butter	Cream	Cheese	Honey
User 1	1	?	2	1	0.5	1
User 2	0.5	?	1	0.5	2	1

In this way, the historical sequential purchase patterns of a user are analyzed and integrated into the user-item matrix input to enhance and improve the rating quality and quantity.

3) Integrating frequency, price factor in recommendation

Traditional CF & CBF techniques, consider only the explicit ratings of an item for providing a recommendation. These methods just use the binary user-item rating purchase matrix which doesn't reflect much regarding how much a user likes an item; how frequently or how long ago a user purchased an item and what quantity of a product was purchased. This information is not integrated into the CF user-rating matrix but can potentially improve the recommendations accuracy and provide effective recommendation to users. In order to handle these challenges, the complex sequential patterns of user purchase behavior such as customer clicks and/or purchases needs to be learned and integrated into the user-item rating matrix input. For example, as seen from the historical purchase data (Table 1.12), a sequential database (Table 1.14) can be created by considering the time period of the purchase which is further mined for frequent sequential purchase patterns by using any SPM algorithm. Similarly, user-item purchase frequency matrix (Table 1.13) can be created from the historical purchase data (Table 1.12), where the values indicates the frequency (number of times an item was purchased) of a user. SPM seeks to explicitly model the timestamps of interactions to explore the influence of different time intervals on next item prediction (Li, Wang & Mcauley, 2020). Most of the sequential pattern-based recommendation systems sort items by interaction timestamps, to predict the next item likely to be interacted with. Now, to understand how SPM integrates the price factor in recommendation:

Consider the purchase sequential database (Table 1.14). The pattern {milk, butter} may be highly frequent but may be uninteresting as it represents a purchase behavior that is common and may yield a low profit. On the other hand, several patterns such as {caviar, champagne} (if exists in the database), may not be frequent but may yield a higher profit. Hence, to find interesting patterns in data, other aspects can be considered such as the profit or utility. This discovery of high utility patterns in databases (Yao & Hamilton, 2006), selects interesting patterns based on minimum utility rather than minimum support. Utility is a quantitative representation of user preference and can be defined as "A measure of how 'useful' (i.e. profitable) an itemset is" (Yao & Hamilton, 2006). In practice, the utility value of a pattern can be measured in terms of cost, profit, aesthetic value, or other measures of user preference. The utility is introduced into pattern mining to mine for patterns of high utility by considering the quality (such as profit) and quantity (such as number of items purchased) of itemsets.

In many real-world applications, user's current interests are intrinsically dynamic and evolving, influenced by their historical behaviors (Sun et al., 2019). Fortunately, the user consumption histories (historical purchase data) offer crucial clues that help us, tackle this important problem. When we visit web pages or purchase things from online stores, we leave a trace of time ordered sequence of items that we have seen or bought. Thus, we can employ a powerful data mining process called SPM, to discover temporal patterns that are frequently repeated among different users, from these historical purchase sequences (Xiao, Liang & Meng, 2019). SPM adds an additional dynamic by taking the order of previous interactions into account (Rendle, Freudenthaler & Schmidt-Thieme, 2010). Thus, if these complex sequential patterns of user purchase behavior are learned and integrated into the user-item rating matrix input, the recommendation quality can be improved in terms of accuracy, sparsity, diversity and novelty.

1.4 Problem Definition

The goal of this thesis is to accentuate the importance of integrating user's sequential purchase behavior (SPM) with the user-item interaction (CF) to improve the quality of recommendations in the application domain of E-commerce, by performing an in-depth comparative review of different Sequential pattern based collaborative E-commerce recommendation systems (SP-based E-commerce RS). The surveyed systems such as ChoRec05, ChenRec09, HuangRec09, LiuRec09, ChoiRec12, Hybrid Model RecSys16, Product RecSys16, SainiRec17, HPCRec18 and HSPCRec19 have attempted to integrate historical purchase sequences or sequential patterns with CF to recommend items to users. The review of these systems involves comparison of their features such as their recommendation accuracy and functionalities (e.g., able to recommend novel or diverse products), recommendation approaches, improving on understanding of the system's algorithms with example application of system to a clear example, highlighting their strengths, weaknesses and future prospects in the recommendation process.

Thus, thesis problem definition can be stated in general terms as follows: Given a set of existing SPM based E-Commerce recommendation systems, compare their (i) features (e.g., recommendation accuracy, their user-rating matrix input data sparsity ratio, ability to recommend novel products, ability to recommend diverse product, ability to scale up to frequently changing products or user (scalability), etc.), (ii) recommendation techniques (e.g., collaborative filtering, content based, knowledge based, hybrid and with algorithmic details), (iii) summary of their algorithmic application to example problem, (iv) strengths, weaknesses and future prospects.

1.5 Thesis Contribution

This thesis will accentuate the importance of sequential patterns in recommendation systems for the application domain of E-commerce by showing that integrating sequential patterns (through sequential pattern mining) into user-item matrix (which is a basis for technique like collaborative filtering method) provides effective recommendations by closing the high sparsity matrix problem and thus, improving the recommendation quality through more novel/diverse recommendations. A deep discussion of SP-based E-commerce RS, their features, solutions, strengths, limitations and prospects for future work are discussed in this thesis to lay the foundation for researchers and practitioners to foster innovations in the area of recommendation systems. The contributions of this thesis to research can be summarized as follows in the next two subsections:

1.5.1 Feature Contributions

After careful investigation of SP-based E-commerce RS, we identified that integrating sequential patterns into user-item matrix in recommendation systems:

(i) Improved the recommendation **accuracy**

(ii) Reduced **data Sparsity**

(iii) Increased **Novelty Rate of recommendations**

(iv) Improved **Scalability of the recommendation system**

Thus, we considered the above performance factors in this thesis to measure the importance of sequential patterns in recommendation systems.

Being able to accurately predict the relevance of items for users is and will be a central problem of recommendation systems research. Increasing the prediction accuracy therefore is a relevant goal of research (Jannach & Jugovac, 2019). Although accuracy metrics are arguably the most important components of the evaluation, they can often provide an incomplete picture of the user experience, as recommending items that the user might have bought anyways might be of little business value. So, focusing on accuracy alone can lead to monotone recommendations and limited discovery. Therefore, it is important to design the evaluation system carefully so that the measured metrics truly reflect the effectiveness of the system from the user perspective. Thus, the experimentation should be able to assess multiple, possibly competing goals in parallel such as Accuracy, Novelty and Scalability.

The problem of Sparsity and Scalability affects efficiency when there is a huge number of items and weights/ratings associated with each item and when a profile is maintained for each user. This problem leads to large matrices of data that require transformation and processing on the fly for online recommendations. Sparsity problem appears when there is a huge matrix with only few weights or ratings, i.e. this problem arises when there are many items to be recommended, but only few recommendations are provided, or recommendations are mostly targeting only a subset of the items. To solve these limitations of recommendation systems, SPM has been used and Chapter 3 provides a detailed description of how SPM solved these problems.

1.5.2 Procedural Contributions

In order to achieve the above discussed functionalities:

(1) A comprehensive survey of SP-based E-commerce RS is carried out with experimental comparison of few surveyed algorithms. The proposed taxonomy for algorithms provides a deep understanding of the different SP-based E-commerce RS, their component features, and the different techniques and methods used in research so far, highlighting comparative advantages and drawbacks of these algorithms.

(2) In addition, we summarized the current research progress in the area of SP-based E-commerce RS from a technical perspective which brings about not only an overview of the progress made so far but also the necessary technical details.

(3) We discussed the challenges and open issues in the existing recommendation systems and identified the new trends and prospects of future directions in this research field to share the vision and expand the horizons of recommendation systems research.

1.6 Discussion

In the light of the problems identified (Sparsity, Scalability, Novelty & Accuracy), we put forth the following research questions to be answered (Chapter 3) by this research:

(1) How has SPM been used to handle sparsity problem, improve recommendation accuracy, novelty and scalability in the reviewed systems?

(2) What are the existing challenges faced by E-commerce recommendation systems and how they can be solved?

(3) What is the importance of SPM in recommendation systems for the application domain of E-commerce?

1.7 Thesis Outline

In this chapter, we have introduced Recommendation systems and its significance in the domain of E-commerce, the classification of recommendation system techniques and their limitations, Sequential Pattern mining and its need in the E-commerce recommendation systems followed by the thesis problem definition and contributions. The remainder of the thesis is organized as follows:

Chapter 2 discusses related work by reviewing significant existing comprehensive survey articles in the field of SPM and Recommendation Systems and gives a brief summary of the SP-based E-commerce RS being surveyed in terms of the four factors listed in the problem definition of section 1.4. That is, the systems are surveyed in terms of their features, recommendation techniques and algorithms, summary of their algorithmic application to sample problem and strengths, weaknesses and prospects.

Chapter 3 discusses the proposed set of factors and features for defining taxonomies of the surveyed related work (i.e. SP-based E-commerce RS).

Chapter 4 discusses the comparative study and experimental performance analysis of few of the surveyed SP-based E-commerce RS.

Chapter 5 concludes the paper and gives a brief outlook on future research directions.

CHAPTER 2: RELATED WORK

Recommendation Systems have evolved into a fundamental tool for helping users make informed decisions and choices, especially in the era of big data in which customers must make choices from a large number of products and services available (Wang, Cao & Wang, 2019). The number of research publications on recommendation systems have increased exponentially in these years, providing strong evidence of the inevitable pervasiveness of recommendation systems research. Although several works have been explored in this research field of recommendation systems applications, the use of sequential patterns for providing recommendations was barely examined. Thus, a comprehensive review and summary of Sequential Pattern-based recommendation systems is required both in academia and in industry for successive researchers and practitioners to better understand the strength and weakness, and application scenarios of these models. As the recommendation systems have been playing a vital and indispensable role in various information access systems to boost business and facilitate decision-making process, and are pervasive across numerous web domains such as e-commerce, this thesis seeks to provide a systematic review and a taxonomy of current research on SP-based E-commerce RS and outline the open challenges in this area. We will summarize all existing significant survey articles in the field of SPM and recommendation systems which constitutes some relevant work in this area to justify the need and use of this survey.

2.1 Survey Articles on Recommendation systems

2.1.1 Towards the next generation of recommender systems: A survey of the state-of-the-art and possible extensions (Adomavicius & Tuzhilin, 2005)

An overview of the field of recommender systems was presented in this work which describes the current generation of recommendation methods and the formulation of recommendation problem was reduced to the problem of estimating ratings for the items that have not been seen by a user. Once the ratings for unrated items were predicted, the item(s) with the highest estimated rating(s) are recommend to the user. Recommender systems are usually classified into the following three main categories based on how recommendations are made (Balabanovic & Shoham, 1997): content-based, collaborative, and hybrid recommendation approaches. In content-based recommendation systems, the user will be recommended items similar to the ones the user preferred in the past. Content-based systems focus on recommending items containing textual information, such as documents, Web sites (URLs), and Usenet news messages. The collaborative filtering systems recommends the user, the items that people with similar tastes and preferences liked in the past. Few examples of collaborative recommender systems include the book recommendation system from Amazon.com, the Jester system that recommends jokes (Goldberg, Roeder, Gupta & Perkins, 2001) and GroupLens (Konstan et al., 1997) system that recommends Usenet news. Hybrid approaches combine collaborative and content-based methods.

Various limitations of these recommendation methods were described such as content-based recommender systems have few limitations which are Limited Content Analysis, Overspecialization and New User Problem whereas collaborative systems face the New User Problem, New Item Problem and Sparsity issues. Possible extensions that can improve recommendation capabilities were discussed to make recommender systems applicable to an even broader range of applications. These extensions include, better methods for representing user behavior and the information about items to be recommended, more advanced recommendation modeling methods, incorporation of various contextual information into the recommendation process, utilization of multicriteria ratings, supporting multidimensionality of recommendations, development of less intrusive and more flexible recommendation methods that also rely on the measures that more effectively determine performance of recommender systems.

Though this survey lays the foundation for many researchers to review the field of Recommendation systems, this work is outdated as there are numerous extensions to the recommendation algorithms and applications with more recent techniques and advances. Also, there are other challenges that have emerged in this area alongside the advancements.

2.1.2 A Taxonomy of Sequential Pattern Mining Algorithms (Mabroukeh & Ezeife, 2010)

A taxonomy for classifying SPM algorithms in the literature with web usage mining as an application, was introduced first by (Mabroukeh & Ezeife, 2010) based on important key features supported by these techniques. This work also attempted to provide a comparative performance analysis of algorithms from each of the categories in the taxonomy and discussed their theoretical aspects by providing a deep discussion of features in each category of algorithms, highlighting weaknesses and techniques that require further research. The proposed taxonomy was composed of four main categories of SPM algorithms, namely, **apriori-based:** AprioriAll (Agrawal & Srikant, 1995), GSP (Agrawal & Srikant, 1996), PSP (Masseglia, Poncelet & Cicchetti, 1999), SPAM (Ayres, Flannick, Gehrke & Yiu, 2002); **pattern-growth:** FreeSpan (Han et al., 2000), PrefixSpan (Pei, Han, Mortazavi & Pinto, 2001), WAP-mine (Pei, Han, Mortazavi & Zhu, 2000), FS-Miner (El-Sayed, Ruiz & Rundensteiner, 2004); **early-pruning:** LAPIN (Yang, Wang & Kitsuregawa, 2007), HVSM (Song, Hu & Jin, 2005), DISC-all (Chiu, Wu & Chen, 2004) and **hybrid** algorithms: SPADE (Zaki, 2001), PLWAP (Ezeife & Lu, 2005). The apriori-based algorithms are based on the rule that "all nonempty subsets of a frequent itemset must also be frequent" also described as anti-monotonic (or downward-closed) property. Major drawback of this type of algorithms is that they require multiple scans of the database which makes them computationally expensive and it also requires a lot of processing time. Pattern growth algorithms typically implement a physical tree data structure representation of the search space in search of frequent sequential patterns, but the trees can grow to be very large and consume a lot of memory. The key idea in early pruning algorithms is the position induction feature i.e. it is very important for an efficient algorithm not to scan the sequence database iteratively. This is achieved by the early pruning algorithms with the use of bitmaps or positional tables. This way, an algorithm utilizes less memory during the mining process, in comparison with tree projections. One disadvantage is that the amount of computation incurred by bitwise (usually AND) operations used to count the support for each candidate sequence. Methods other than tree projection should be investigated further for finding reliable SPM techniques. Hybrid algorithms combine several features that are characteristics of more than one of the three categories (apriori-based, pattern-growth and early-pruning algorithms).

The quest for finding a reliable SPM algorithm was achieved by this work after a careful investigation of the SPM algorithms available in the literature and shows that the important heuristics employed include the following:

- using optimally sized data structure representations of the sequence database
- early pruning of candidate sequences
- mechanisms to minimize support counting and maintaining a narrow search space.

Though there are various other surveys (Zhao & Bhowmick, 2003; Han et al., 2007; Pei et al., 2007) that have been published in the area of sequential pattern mining, (Mabroukeh & Ezeife, 2010) laid the foundation by proposing a taxonomy that presents a hierarchical, tabular, and a chronological ordering of the sequential pattern-mining algorithms along with their features and the supporting theory which each algorithm in the taxonomy is based on. We derived the motivation for conducting a study on sequential pattern-based algorithms from this work, as sequential pattern mining is a very active research topic, where hundreds of papers present new algorithms and applications every year, including numerous extensions of sequential pattern mining for specific needs. Though the reviewed algorithms were discussed in detail and presented with running examples, this work is no longer up to date as it does not discuss the most recent techniques, advances and challenges in the field. The survey was directed only towards the general case of sequential pattern mining, and did not consider algorithms specific to closed, maximal or incremental sequences, neither did it investigate special cases of constrained, approximate or near-match sequential pattern mining.

2.1.3 Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey (Bokde, Girase & Mukhopadhyay, 2015)

Collaborative Filtering (CF) algorithms are most commonly used in Recommendation Systems. User's preferences for items are stored in the form of ratings matrix, which are used to build the relation between users and items to find user's relevant items. In the past decades due to the rapid growth of Internet usage, vast amount of data is generated, and thus, CF algorithms faces issues with sparsity of ratings matrix (for each user only a relatively small number of items are rated) and growing nature of data (size of processed datasets). This work studied various Matrix Factorization models such as Singular Value Decomposition (SVD), Principal Component Analysis (PCA), Probabilistic Matrix Factorization (PMF) and Non-Negative Matrix Factorization (NMF) to deal with these challenges of CF algorithms and attempts to present a comprehensive survey, which can be served as a roadmap for research and practice in this area.

Matrix Factorization (MF) is a powerful technique to find the hidden structure behind the data and is an unsupervised learning method for latent variable decomposition and dimensionality reduction, applied successfully in spectral data analysis and text mining. Mostly, MF models are based on the latent factor model, which characterizes both items and users by vectors of factors inferred from the items rating patterns. The high correspondence between user factors and item factors leads to a recommendation. MF methods have become popular recently by combining good scalability with predictive accuracy and offers much flexibility for modeling various real-life applications. MF models map both users and items to a joint latent factor space of dimensionality 'f' and the user-item interactions are modeled as inner products in that space, i.e. modeled as the product of a user factor matrix and an item factor matrix.

Singular Value Decomposition (SVD), reduces the dimensionality of the ratings matrix and identifies latent factors in the data. Applying SVD in the domain of CF requires factoring the user-item rating matrix, and this study shows that SVD was able to handle massive dataset, sparseness of ratings matrix, scalability and cold-start problem of user-based/item-based CF algorithms efficiently. Use of SVD model in CF algorithms increases computation cost and finding a lower dimensional feature space is a key issue in an SVD decomposition. To overcome this problem researchers have come up with Stochastic Singular Value Decomposition (SSVD) MF model, which not only reduces the computation cost of CF algorithms but also increases the accuracy, preciseness and efficiency of the CF algorithms.

Like SVD, PCA reduces dimensionality of matrix by optimally projecting highly correlated data along a smaller number of orthogonal dimensions. PCA is a classical statistical method to find patterns in high dimensionality data sets. PMF is a probabilistic linear model with Gaussian observation noise which models the user preference matrix as a product of two lower-rank user and item matrices, whereas, NMF is a group of algorithms in multivariate analysis and linear algebra where a matrix X is factorized into two matrices P and Q, with the property that all three matrices have no negative elements.

The authors have provided an overview on MF models such as SVD, PCA, PMF and NMF but haven't dealt with any of the papers that used these algorithms to solve the CF method issues like sparsity and increasing dataset size. Despite the claim of authors regarding discussing the MF models, there are no mathematical or algorithmic details specified for PCA and NMF models. Also, this is a very limited work in the area of CF algorithms as there are many other MF models which weren't explored in this survey such as Funk MF, Hybrid MF, SVD++, Asymmetric SVD etc. for the purpose of decomposing the user-item interaction matrix into the product of two lower dimensionality matrices.

2.1.4 Sequence-Aware Recommender Systems (Quadrana, cremonesi & Jannach, 2018)

This work (Quadrana, cremonesi & Jannach, 2018) reviewed the existing literature on sequence-aware recommender systems that consider information from sequentially ordered (and time-stamped) user-item interaction logs as an input to base their recommendations, at least partially, on the sequential patterns that they extract from the data. The interaction logs consist of set of user actions (such as clickstream data) which are usually connected with items like Item purchase/consumption, item view, add-to-catalog, add-to-wish-list, etc. The output of a sequence-aware recommender is a ranked list of item suggestions i.e. one (or more) ordered list of items which can have different interpretations based on goal, domain and application scenario. For example, the output can be of the forms:

1) Usual item-ranking tasks
   - list of alternatives for a given item
   - complements or Accessories
2) Suggested sequence of actions
   - next-track music recommendations
3) Strict sequence of actions
   - course learning recommendations

Based on this review, a categorization of the corresponding recommendation tasks and goals were proposed, and the existing algorithmic solutions were summarized along with the discussion of methodological approaches and the open challenges in this area. Sequence-aware recommenders are typically designed to support certain types of goals and recommendation purposes such as Context adaptation, Trend detection, Repeated recommendation and Consideration of order constraints and sequential patterns in different application scenarios.

This work identified three main classes of algorithms: Sequence learning, Sequence-aware matrix factorization, and Hybrid method in the literature for the extraction of patterns from the sequential log of user actions. Finally, the authors have identified and discussed some of the open research directions in the topics: Intent Detection, Combining Short-Term and Long-Term Profiles, Leveraging Additional Data and General Trends and Towards standardized & more comprehensive evaluations.

The authors have provided an overview on many papers according to their categorization scheme but haven’t dealt in detail with any of the algorithms from the classes identified. This work investigated sequence-aware recommenders in several application domains like E-commerce, Music, POI, App recommendations and web navigation prediction and failed to focus on a single application domain as a complete solution and target the development of algorithms for specific domains. The evaluation approaches for Sequence-Aware Recommender Systems were discussed in general without providing any comparative performance analysis of the techniques.

2.1.5 Other significant work

Recommendation Systems have become increasingly popular in academic research and practical applications. Plenty of research has been done in this field and several surveys on recommendation systems have also been presented. For example, (Burke, 2002) proposed a comprehensive survey on hybrid recommendation systems; (Su & Taghi, 2009) presented a systematic review on CF techniques. However, in recent years, with the constant advent of novel research works in the area of recommendation systems, Deep Learning (DL) has gained the popularity and potential and thus, has been reviewed extensively for the better understanding of this research field. For DL-based recommendations, (Singhal, Sinha & Pant, 2017) summarized DL-based recommendation systems and categorized them into three types: CF, CBF and hybrid ones. (Quadrana, Cremonesi & Jannach, 2018) proposed a categorization of the recommendation tasks and goals and summarized existing solutions. (Batmaz et al., 2018) classified and summarized the DL-based recommendation from the perspectives of DL techniques and recommendation issues and gave a brief introduction of the session-based recommendations. (Xu, Liu & Xu, 2019) divided the existing sequential recommendation methods into Markov-chain, Neural model and Attention mechanism-based recommendation systems. (Wang, Cao & Yan, 2019) illustrated the value and significance of the session-based recommender systems (SBRS) and proposed a hierarchical framework to categorize issues and methods including some DL-based ones. (Zhang et al., 2019) further discussed the state-of-the-art DL-based recommender systems, including several RNN-based sequential recommendation algorithms.

Although these works explored the recommendation systems applications, however, none of these surveys focused on the emphasis of sequential patterns for recommendation systems for the application domain of E-commerce.

2.2 Need for a Sequential Pattern Based Recommendation Systems Survey

With the rapid growth of online information sources and e-commerce businesses, users increasingly need reliable recommendation systems, to highlight relevant items, i.e., next items that the users would most probably like. Over the past two decades, a large amount of research effort has been devoted for developing algorithms that generate recommendations. The most common type of such recommendation system technique is Collaborative Filtering (CF), which takes user's interest in an item (explicit rating) as input in a matrix known as the user-item rating matrix, and produces an output consisting of unknown ratings of users for items from which top N recommended items for target users or top N recommended users for target items are defined. These traditional recommendation systems have some drawbacks. A critical one is that they only focus on a user's long-term static preference while ignoring his or her short-term transactional patterns, which results in missing the user's preference shift through the time. This is an important attribute as the time interval between item purchases is useful to learn the next items for purchase by users. Thus, a powerful data mining process called sequential pattern mining (SPM) is employed to discover temporal patterns that are frequently repeated among different users, from the historical purchase sequences. SPM adds an additional dynamic by taking the order of previous interactions into account. The modeling of these third order interactions (between a user, an item under consideration and the previous item consumed) facilitates a more engaging user experience, resulting in recommendations that are more responsive to recent user and item dynamics (Shani, David & Ronen, 2002). Sequential Pattern-based recommendation systems treat the user-item interactions as a dynamic sequence and take the sequential dependencies into account to capture the current and recent preference of a user for more accurate, customized and dynamic recommendations.

SPM techniques, therefore, have been used recently (Yap, Li & Yu, 2012) to make the recommendations more effective and the accuracy of recommendation systems will be improved if these complex sequential patterns of user purchase behavior are learned and integrated into the user-item rating matrix, as the input becomes more informative before it is fed to CF. Thus, integrating CF and SPM improves recommendation accuracy, diversity and quality and is the focus of this survey. However, to the best of our knowledge, none of the aforesaid surveys focused on the emphasis of sequential patterns for recommendation systems in the application domain of E-commerce which laid out the motivation for conducting a study on sequential pattern-based e-commerce recommendation systems. This is a novel research work investigated in recent years in the relevant communities and the importance of sequential patterns in recommendation systems is accentuated through this research for the application domain of E-commerce. The goal of this survey is to thoroughly review literature to systematically summarize and explore SP-based E-commerce RS, thus providing a foundation and a comprehensive view with a rich list of relevant resources for the community to identify open problems currently limiting real-world implementations and to point out future directions along this dimension.

Now, let us get an insight into how SPM algorithms work with an example in the next section followed by a summary of the SP-based E-commerce RS being surveyed.

2.3 Sequential Pattern Mining Algorithms

In this section, we discuss the SPM algorithms that our surveyed systems used for the purpose of recommendations.

2.3.1 GSP (Generalized sequential pattern mining) algorithm (Agrawal & Srikant, 1996)

GSP algorithm (Generalized Sequential Pattern algorithm) is one of the first algorithm for discovering sequential patterns in sequence databases, proposed by (Agrawal & Srikant, 1996). It uses an Apriori-like approach for discovering sequential patterns. The algorithms for solving sequence mining problems are mostly based on a priori (level-wise) algorithm. One way to use the level-wise paradigm is to first discover all the frequent items in a level-wise fashion. It simply means counting the occurrences of all singleton elements in the database. Then, the transactions are filtered by removing the non-frequent items. At the end of this step, each transaction consists of only the frequent elements it originally contained. This modified database becomes an input to the GSP algorithm. GSP algorithm makes multiple database passes. In the first pass, all single items (1-sequences) are counted. From the frequent items, a set of candidate 2-sequences are formed, and another pass is made to identify their frequency. The frequent 2-sequences are used to generate the candidate 3-sequences, and this process is repeated until no more frequent sequences are found. There are two main steps in the algorithm.

- **Candidate Generation:** Given the set of (k-1)-frequent sequences F_{k-1}, the candidates for the next pass are generated by joining $F_{(k-1)}$ with itself. A pruning phase eliminates any sequence at least one of whose sub sequences is not frequent.
- **Support Counting:** Normally, a hash tree–based search is employed for efficient support counting. Finally, non-maximal frequent sequences are removed.

Let us, consider daily sequential database (Table 2.1) as input, minimum support =2 and candidate set (C1) = {A, B, C, D, E, F, G}.

Step 1: Find 1- frequent sequence (L_1) to keep only sequence with occurrence or support count in the database greater than or equal to minimum support. For example, L_1= {<(A):4>, <(B):5>, <(C):3>, <(D):4>, <(F):4>, <(G):4>}.

Table 2.1 Sequence database representing customer purchase

SID	Sequences
1	<(A), (B), (FG), (C), (D)>
2	<(B), (G), (D)>
3	<(B), (F), (G), (A, B)>
4	<(F), (A, B), (C), (D)>
5	<(A), (B, C), (G), (F), (D, E)>

Step 2: Generate candidate sequence ($C_{k=2}$) using $L_1 \bowtie_{GSPJoin} L_1$. To generate larger candidate set 2, use 1-frequent sequences found in step 1, which can be written as $L_{(k-1)} \bowtie_{GSPJoin} L_{(k-1)}$ and it requires every sequence (W_1) found in first $L_{(k-1)}$ joins with other sequence (W_2) in the second, if subsequences obtained by removal of first element of W_1 and last element of W_2 are same.

Step 3: Find 2- frequent sequences (L_2) by counting occurrence of 2-sequences in candidate sequence (C_2) to keep only sequence with occurrence or support count in the database greater than or equal to minimum support.

Step 4: Generate candidate sequence ($C_{k=3}$) using $L_2 \bowtie_{GSPJoin} L_2$

Step 5: Find 3- frequent sequences (L_3) to keep sequences with occurrence or support count in the database greater than or equal to minimum support.

Step 6: Repeat process of candidate generation and pruning until result of candidate generate (C_k) and prune (L_k) for finding frequent sequence is an empty set.

Output: Finally, the output frequent sequences are union of $L_1 \cup L_2 \cup L_3 \cup L_4$

Table 2.2 n-frequent sequences generated by GSP algorithm

1-Frequent Sequences	**2-Frequent Sequences**	**3-Frequent Sequences**	**4-Frequent Sequences**
<(A)>, <(B)>, <(C)>, <(D)>, <(F)>, <(G)>	<(A), (B)>, < (A, B)>, <(A),(C)>, <(A), (D)>, <(A),(F)>, <(A), (G)>, <(B),(C)>, <(B), (D)>, <(B), (F)>, <(B), (G)>, <(C),(D)>, <(F), (A)>, <(F), (B)>, <(F), (C)>, <(F),(D)>, <(G), (D)>	<(F), (C), (D)> <(B), (G), (D)> <(B), (F), (D)> <(B), (C), (D)> <(A), (G), (D)> <(A), (F), (D)> <(A), (C), (D)> <(A), (B), (G)> <(A), (B), (F)> <(A), (B), (D)>	<(A), (B), (G), (D)> <(A), (B), (F), (D)>

2.3.2 FreeSpan (Frequent Pattern-Projected Sequential Pattern Mining) algorithm (Han et al., 2000)

FreeSpan stands for Frequent Pattern-Projected Sequential Pattern Mining and starts by creating a list of frequent 1-sequences from the sequence database called the frequent item list (f-list), it then constructs a lower triangular matrix of the items in this list. This matrix contains information about the support count of every 2-sequence candidate sequence that can be generated using items in the f-list and is called S-Matrix. For a sequential pattern α from S-Matrix, the α-projected database is considered the collection of frequent sequences having α as a subsequence. Infrequent items and items following those infrequent items in α are ignored. For the next step a table is constructed with length 2-sequences (i.e., frequent 2-sequences) along with annotations on repeating items and annotations on projected databases that help in locating these projected databases in the third and last scans of the database without referring to the S-Matrix. The S-Matrix is now discarded, and mining picks up using the projected databases.

Let us, consider a sequence database (Table 2.3) as input and minimum support=25%

Table 2.3 Sequence Database D

Sequence ID for each Customer	Data Sequence
1	<(a)(e)>
2	<(fg)a(fbkc)>
3	<(ahcd)>
4	<a(abcd)e>
5	<e>

Step 1: The first scan of D generates the list of frequent 1-sequences as f-list = {a:4, b:2, c:3, d:2, e:3}, thus the complete set of sequential patterns can be divided into 5 disjoint subsets, each of which has its own projected database and is mined in a divide-and conquer method.

Step 2: Now construct the S-Matrix to count the support of each 2-sequence, as follows. Consider the f-list $\{i_1, i_2, . . . , i_n\}$, F is a triangular matrix F[j, k] where$1 \le j \le m$ and $1 \le k \le j$, such that m is the number of items in the itemset of the sequence under consideration. F[j, j] has only one counter, which is the support count for 2-sequence $<i_j i_j>$. Every other entry has three counters (A, B, C); A is the number of occurrences of $<i_j i_k>$, B is the number of occurrences of $<i_k i_j>$, and C is the number of occurrences in which i_k occurs concurrently with i_j as an itemset $<(i_j\ i_k)>$. The S-Matrix for the 2-sequences of Table 2.3 can be seen in Table 2.4, which is filled up during a second scan of the database D.

Table 2.4 S-Matrix for constructing 2-sequences from Sequence Database D

a	1				
c	(2,0,2)	0			
e	(2,0,0)	(1,0,0)	0		
b	(2,0,1)	(0,0,0)	(0,1,0)	0	
d	(1,0,2)	(0,0,2)	(0,1,0)	(0,0,1)	0
	a	**c**	**e**	**b**	**d**

For example, the entry (2,0,2) in the second row, first column of the matrix in Table 2.4, means that the sequence <ac> occurs 2 times, the sequence <ca> occurs zero times, and the sequence <(ac)> occurs 2 times in D.

Step 3: The third step builds level-2-sequences from the candidate sequences in the S-Matrix and finds annotations for repeating items and projected databases in order to discard the matrix and generate level-3 projected databases Table 2.5, built from parsing the matrix row-wise, bottom-up.

Table 2.5 Pattern generation from S-Matrix

Item	**Frequent 2-Sequences**	**Annotations on Repeating Items**	**Annotations on Projected DBs**
d	<(ad)>:2, <(cd)>:2	<(ad)>, <(cd)>	<(ad)>:{c}
b	<ab>:2	<ab>	-
e	<ae>:2	<ae>	-
c	<ac>:2, <(ac)>:2	{ac}	-
a	-	-	-

Consider the row for d, since, F[a, d], F[c, d] and F[a, c] form a pattern generating triple and F[a, d] = (1,0,2), meaning only <(ad)> is valid (because its support count is above the threshold), the annotation for the projected database should be <(ad)> :{c}, which indicates generating <(ad)>-projected database, with c as the only additional item included.

Step 4: From the generated annotations, scan the database one more time to generate item repeating patterns {(ad):2, (cd):2, ab:2, ae:2, ac:2, (ac):2}. There is only one projected database <(ad)>:{c} whose annotation contains exactly 3 items, and its associated sequential pattern is obtained by a simple scan of the projected database. If it contains more than 3 items, S-Matrix is constructed for this projected database and recursively mines it for sequential patterns the same way.

FreeSpan examines substantially fewer combinations of subsequences and runs faster than GSP, due to the process of pruning candidate sequences in the S-Matrix before they are generated, especially when the dataset grows larger. The major cost of FreeSpan is the computation and creation of the projected databases. If a pattern appears in each sequence of a database, its projected database does not shrink. Also, since a k-subsequence may grow at any position, the search for a length (k + 1) candidate sequence needs to check every possible combination, which is costly.

2.3.3 PrefixSpan (Prefix-projected sequential pattern mining) algorithm (Pei, Han, Mortazavi & Pinto, 2001)

PrefixSpan (Prefix-projected sequential pattern mining) examines only the prefix subsequences and projects only their corresponding postfix subsequences into projected databases. This way, sequential patterns are grown in each projected database by exploring only local frequent sequences. To illustrate the idea of projected databases, consider <f>, <(fg)>, <(fg)a> which are all prefixes of sequence <(fg)a(fbkc)> from Table 2.3, but neither <fa> nor <ga> is considered a prefix. On the other hand, <(g)a(fbkc)> is the postfix of the same sequence with respect to <f>, and <a(fbkc)> is the postfix with respect to prefix <(fg)>. A running example of PrefixSpan on database D (Table 2.3) acts in three steps:

Step 1: Find all 1-itemset sequential patterns by scanning the database D. We get {a:4, b:2, c:3, d:2, e:3} along with their support counts.

Step 2: Divide the search space to get projected databases like FreeSpan. This example generates 5 disjoint subsets according to the 5 prefixes <a>, < b>, <c>, <d>, <e>.

Step 3: Find subsets of sequential patterns; these subsets can be mined by constructing projected databases, like FreeSpan, and mining each one recursively.

To find sequential patterns having prefix <a>, we extend it by adding one item at a time. To add the next item x, there are two possibilities (Liu, 2007):

(1) the algorithm joins the last itemset of the prefix (i.e., <(ax)>) and

(2) it forms a separate itemset (i.e., <ax>)

So, to produce <a> - projected database: if a sequence contains item <a>, then the suffix following the first <a> is extracted as a sequence in the projected database. The second sequence (second row) of Table 2.3, <(fg)a(fbkc)>, is projected to <(bc)> where f and k are removed because they are infrequent. The third sequence is projected to <(cd)>, the fourth to <(abcd)e>, eventually the final projected database for prefix <a> contains the following sequences: e, (bc), (cd), (abcd)e; similarly all the other prefixes are given in Table 2.6.

Table 2.6 PrefixSpan on Sequence Database D

Prefix	**Projected Database**	**Sequential Patterns**
<a>	<e>, <(bc)>, <(_cd)>, <(abcd)e>	a, ab, ac, (ac), (ad), ae
<b>	<(_c)>, <(_cd)e>	b, (bc)
<c>	<(_d)>, <(_d)e>	(cd)
<d>	<e>	-
<e>	-	-

Now we need to find all frequent sequences of the form <(ax)>, two templates are used: <(_x)> and <ax> to match each projected sequence to accumulate the support count for each possible x (x matches any item). The second template uses the last itemset in the prefix rather than only its last item. In the example here, they are the same because there is only one item in the last itemset of the prefix. Then, we need to find all frequent sequences of the form <ax>; in this case, xs are frequent items in the projected database that are not in the same itemset as the last item of the prefix. Table 2.6 contains all the frequent sequential patterns generated for this example using PrefixSpan. Looking at the patterns generated for prefix <a>, after finding the frequent 2-sequences (namely, ab, ac, (ac), (ad), ae), we recursively create projected databases for them and start mining for frequent 3-sequences (the example here does not have any).

The key advantage of PrefixSpan is that it does not generate any candidates. It only counts the frequency of local items. It utilizes a divide-and-conquer framework by creating subsets of sequential patterns (i.e., projected databases) that can be further divided when necessary. PrefixSpan performs much better than both GSP and FreeSpan. The major cost of PrefixSpan is the construction of projected databases. To further improve mining efficiency, bilevel projection and pseudo-projection can be used. Bilevel projection uses the S-Matrix, introduced in FreeSpan (Han et al., 2000), instead of the projected databases. It contains support counters as well as all 1-sequences. Support counters in the S-Matrix tell which 3-sequences can be generated and which not, in order for the algorithm to search for them in the database. The authors refer to this as 3-way apriori-checking. Pseudo-projection is used when PrefixSpan runs only in main memory i.e. disk-based processing is not allowed. That is, instead of creating physical projected databases in memory, pointers or pointer objects can be used, such that every projection contains two pieces of information: pointer to the sequence in the database and offset of the postfix in the sequence for each size of projection.

2.4 Existing Sequential pattern-based E-commerce Recommendation Systems

The main aim of e-commerce websites is to turn their visitors into customers. For this purpose, recommendation system is used as a tool that helps in turning clicks into purchases. Obtaining explicit ratings often faces problems such as authenticity of the ratings given by customers/users and unwillingness of users in providing ratings to the items. Thus, implicit ratings play a vital role in providing refined ranking of products. Preference level of the customers are predicted based on CF approach using implicit details like purchase history, browsing history, search patterns, time spent on specific pages and mining click stream paths of like-minded users. As the transaction data provides sets of preferred items and can be used to predict future customer preferences, some researchers applied the association rule mining technique to extract the sequences to improve performance of recommendation systems (Chun, Oh, Kwon & Kim, 2004; Kazienko & Pilarczyk, 2008). However, such systems incorporate customer transaction data from only a single temporal period, which omits the dynamic nature of a customer's access sequences. Unlike association rules, sequential patterns (Mooney & Roddick, 2013) may suggest that a user who accesses a new item in the current time period is likely to access another item in the next time period. Thus, SPM techniques have been used for extracting the complex sequential patterns of user purchase behavior and if these patterns are learned and included in the user-item matrix input, accuracy of the recommendation system will be improved as the input becomes more informative before it is fed to CF. Thus, integrating CF and SPM of historical purchase data will improve the recommendation quality, reduce the data sparsity and increase the novelty of recommendations.

Existing E-commerce recommendation systems that can be found in the literature, which have combined CF with some form of historical purchase sequences (SPM) to recommend items to users are those referred to as Model Based Approach (Cho & Kim, 2005), Pattern Segmentation Framework (Chen, Kuo, Wu & Tang, 2009), Sequential pattern based collaborative recommender system (Huang et al., 2009), Segmentation based approach - LiuRec09 (Liu, Lai, & Lee, 2009), Hybrid Online Product recommendation – ChoiRec12 (Choi, Yoo, Kim & Suh, 2012), Hybrid Model - HM (Fang, Zhang & Chen, 2016), Product Recommendation System – PRS (Jamali & Navaei, 2016), Sequential pattern based recommender system – SainiRec17 (Saini, Saumya & Singh, 2017) Historical clickstream-based recommendation - HPCRec18 (Xiao & Ezeife, 2018) and Historical Sequential Recommendation - HSPCRec18 (Bhatta, Ezeife & Butt, 2019). A brief overview of these surveyed systems is provided below.

2.4.1 A hybrid of association rule mining and collaborative filtering for product recommendation by Cho, Cho & Kim, 2005 (ChoRec05)

A **hybrid recommendation system** that **combines SOM clustering & Association rule based sequential cluster rules** was proposed for mining the changes in customer buying behavior over time. The recommendation procedure is divided into two components called a model building phase and a recommendation phase. A model-building phase is performed once to create a reliable model from the customer transaction database and includes the following steps: transaction clustering using SOM clustering technique, identifying cluster sequences, extracting the cluster sequential rules using association rule mining. The recommendation phase is divided into the following three steps: determining cluster sequences, matching the cluster sequence such that a target customer's purchase sequence is compared with the purchase sequence stored in the association rule base and finally a set of products that the target customer is most likely to purchase are generated by selecting the top N most commonly purchased products in the cluster.

Model-building phase

Transaction clustering: SOM clustering technique was used to obtain transaction clusters. A self-organizing map (SOM) is a type of artificial neural network (ANN) that is trained using unsupervised learning and a method to do dimensionality reduction. As the number of products can be in tens of thousands in a retail business, the number of dimensions would be increased with the increase in number of products. Thus, this approach suggests the use of a product taxonomy which provides an effective dimensionality reduction while improving clustering results. Product taxonomy represents the hierarchical relationships among products as the domain specific knowledge of marketing managers or domain experts. The transactions are transformed into an input matrix composed of a bit vector, and these time-ordered vectors for a customer represents the purchase history of the customer and this input matrix can be thought of as the dynamic profile of the customer.

Identification of cluster sequences: The transaction clustering results in each cluster representing only a group of transactions with similar patterns. These clusters are rearranged by customer and by time period for the identification of dynamic behavior of each customer. The cluster sequence of a customer is learned by identifying the cluster to which each transaction of the customer belongs, during each time period, i.e. if L_i is the behavior locus of customer i, then, the behavior locus L_i is identical to the changes in the cluster number of customer i during l periods.

Extraction of sequential cluster rules: The cluster of a target customer at time T is discovered based on their past behavior. To mine customer behavior according to purchase time, association rule R_j is adopted for determining the most frequent pattern with confidence.

Equation 2.1: Association rule to mine customer behavior in ChoRec05

$$R_j = r_{j,\,T-l+1,}\ \ldots\ldots\ldots\ r_{j,\,T-1} \rightarrow r_{j,T}\,(Support_j, Confidence_j)$$

where rule R_j indicates that, if the locus of a customer is $r_{j,\,T-l+1,}\ \ldots\ldots\ldots\ r_{j,\,T-1}$, then the behavior cluster for that customer is $r_{j,T}$ at time T.

For doing this, the input data is divided into a conditional part and a consequential part. The conditional part of the association rule is composed of the left-hand-side $< C_{i,\ T-l+1,}\ ....,\ C_{i,\ T-1} >$ and $C_{i,T}$ is assigned to the consequential part.

Recommendation phase

In this phase, given the target customers, the products that are best matched to the dynamic behaviors of these customers are found and the target customer's transactions are converted into behavior locus using the SOM clustering model, as in the previous phase. Finally, the best-matching loci stored in the association rule base are extracted and the top N items are recommended to the target customer.

Determination and matching of the cluster sequences of target customers: Behavior locus prediction begins when a target customer's transactions are entered into the SOM model. It is necessary to know the degree to which the behavior locus of a target customer during *l-1* periods before T is similar to the rules of the association rule base. The cluster locus transformed via the SOM model of a target customer is compared with the association rules derived from other customer's loci, and then the best-matching locus is determined. The degree of correspondence between the association rules in the model base and the behavior locus of a target customer is calculated. The degree of similarity between the two, or the extent to which the behavior locus of a target customer is identical to the conditional part of the association rule in the model base, in the same period, can be used as the correspondence measure.

Recommendation of the top N items: The final step involves the derivation of the top N recommendations from the predicted cluster for a target customer at time T. It can be determined that the top-N product recommendation list for a target customer is the most frequently purchased products from among the products in the cluster.

2.4.2 A sequential pattern-based recommender system for analyzing customer purchase behavior by Chen, Kuo, Wu & Tang, 2009 (ChenRec09)

This study proposed **a sequential pattern-based recommender system** that incorporates RFM (Recency, Frequency and Monetary) concept, where "Recency" represents the length of a time period since the last purchase; a lower value corresponds to a higher probability that the customer will make repeat purchases. "Frequency" denotes the number of purchases within a specified time period; a higher frequency indicates stronger customer loyalty. “Monetary” means the amount of money spent in this specified time period; if a customer has a higher monetary value, the company should focus more resources on retaining that customer. RFM sequential patterns are then defined and a novel algorithm, named RFM-Apriori is developed, for generating all RFM sequential patterns from customer’s purchasing data. The algorithm was developed by modifying the well-known Apriori (GSP) algorithm (Agrawal & Srikant, 1996) and using this algorithm, a pattern segmentation framework was proposed, which allows to partition the RFM-patterns into segments relevant to the RFM criteria, to generate valuable information on customer purchasing behavior for managerial decision-making. By partitioning the patterns into groups based on the RFM indices, a retailer can further compare, contrast, and aggregate these groups of patterns to find possible changes in purchasing patterns over time.

2.4.3 A hybrid of sequential pattern based collaborative recommender system for E-commerce recommendation by Huang et al., 2009 (HuangRec09)

This study proposed a **hybrid recommendation system which is a sequential pattern based collaborative recommender system** that predicts the customer’s time-variant purchase behavior in an e-commerce environment where the customer’s purchase patterns may change gradually. A two-stage recommendation process is developed to predict customer purchase behavior for the product categories, as well as for product items. The time window weight is introduced to provide higher importance on the sequential patterns closer to the current time period that possess a larger impact on the prediction than patterns relatively far from the current time period. Given all of the target customer’s transactional sequences in the current time period T and the previous r periods, T-1, T-2,. . ., and T-r, this study determines the active customer’s most likely purchase items in the next time period T + 1 (target prediction period). The proposed system consists of model training for the target customers and model use (implementation) for the active customers. Active customers are selected from the target customer to receive recommendations during model use. The steps in each of these modules are discussed below.

Model training for the target customers

Identifying the target customers: The target customers can be identified according to customer behavioral variables such as recency, frequency and monetary expenditure (RFM model) (Kaymak, 2001).

Building the dynamic customer profile: Dynamic customer buying behaviors can be modeled by analyzing the customer's periodic transaction data. Given a set of products, PRODUCT = $\{product_i : i = 1, 2, \ldots, N\}$, where N is the number of products, the dynamic customer profile for customer$_c$ is defined as follows:

Equation 2.2: Dynamic customer profile for customer$_c$ in Huang et al., 2009

$$PROFILE^{customer_c} = \{quantity_{period_i}^{product_i} : i = 1,2, \ldots . N; t = T - 0, T - 1, \ldots \ldots T - r\}$$

where $quantity_{period_i}^{product_i}$ is the quantity of $product_i$ that $customer_c$ purchased during $period_i$, T is the current period and r is the number of previous periods considered.

Clustering the customers: The customers are clustered based on their dynamic customer profiles using the GA-based clustering approach, in which a chromosome is a solution for a combination of cluster centers. Thus, the length of a chromosome is equal to the dimensions of a dynamic customer profile multiplied by the number of clusters. The solution quality of a chromosome is measured by the fitness function. The fitter chromosome has higher probability to be selected into the recombination pool using the roulette wheel selection method. The fitness function, used to evaluate the quality of clustering for a chromosome, is defined as

Equation 2.3: Fitness function to evaluate the quality of clustering for a chromosome

$$FITNESS = \sum_{i=1}^{N_c} \sum_{point_j \in cluster_i} DISTANCE(point_j, cluster_i)$$

where N*c* is the number of clusters, $\sum_{point_j \in cluster_i} DISTANCE(point_j, cluster_i)$ is the summation of all pair-wise distances from point j in the $cluster_i$ to the cluster center $center_i$.

Sequential pattern mining for each cluster: A cluster's sequential patterns represent the buying behavior of the customers in that cluster. The proposed sequential pattern-based prediction on the product categories has the following two steps.

Step 1: Generate the customer purchase sequence for each customer: The purchase sequence for a customer in a certain time period is a series of transactions that contain several product categories and is prepared by sorting his/her transactions in each time period according to the transaction date order.

Step 2: Discover the sequential patterns for each cluster: The sequential patterns of each cluster are mined in each time period under a predefined minimum support threshold using any SPM algorithm like GSP (Agrawal & Srikant, 1996), PrefixSpan (Pei et al., 2001) etc.

Model use for the active customer

Assign a proper cluster to the active customer: An active customer is defined as one that receives recommendations from the trained CF recommendation model. Based on the dynamic customer profile, the cluster that an active customer belongs to is determined by calculating the Euclidian distances between the active customer and the cluster centers. A two-stage recommendation process is followed by the cluster selection for the active customer. The two-stage process includes predicting the top-M product categories and recommending the top-N product items.

Top-M product categories prediction: To predict (recommend) the top-M product categories for the active customer based on the sequential patterns: First, generate the candidate sequences (CSs) for the active customer. Then, find the predicted categories by matching candidate sequences with sequential patterns and calculate the total support for the predicted category by summing support values of the matched sequential patterns. Finally, predict the Top-M product categories. The top-M product categories are recommended based on the value of product Category Recommendation Score (CRS). The CRS for the predicted category$_i$ is calculated as follows:

Equation 2.4: Category Recommendation Score for the predicted category$_i$ in Huang et al., 2009

$$CRS^{category_i} = \sum_{\substack{period_t \\ = T-0, T-1, \ldots T-r}} (CATEGORY\_SUPPORT_{period_t}^{category_j} \times WEIGHT_{period_t})\ t$$

where $WEIGHT_{period_t}$ is the time window weight in $period_t$

Top-N product items recommendation: The possible top-N items that the active customer will probably purchase in the target period are generated by calculating the recommendation score for each item in the top-M product categories. The Item Recommendation Score (IRS) for an item among the top-M product categories is calculated as follows.

Equation 2.5: Category Recommendation Score for the predicted category$_i$ in Huang et al., 2009

$$IRS^{item_j} = \sum_{\substack{period_t \\ = T-0, T-1, \ldots T-r}} (PURCHSASE\_FREQUENCY_{period_t}^{item_j} \times WEIGHT_{period_t})\ t$$

where, $PURCHSASE\_FREQUENCY_{period_t}^{item_j}$ is the frequency of $item_j$ bought by all customers in the same cluster in $period_t$. The $PURCHSASE\_FREQUENCY$ is defined as the ‘‘number of times” instead of ‘‘quantity” that customers brought during a certain period.

The top-N items with larger recommendation scores, excluding items that have been bought by the active customer before are recommended to the active customer.

2.4.4 A hybrid of sequential rules and collaborative filtering for product recommendation by Liu, Lai & Lee, 2009 (LiuRec09)

A **hybrid recommendation system which combines segmentation-based sequential rule method with the segmentation-based KNN-CF method** was proposed in this study. Customers are clustered into groups using Recency, Frequency, Monetary (RFM) segmentation with K-means clustering method. Once the RFM segmentation is created, users are further segmented using transaction matrix. The transaction matrix captures the list of items purchased or not purchased by users over a monthly period in a given products list and is used to derive transaction clusters with the use of self-organizing map (SOM) clustering technique (which is a type of artificial neural network (ANN) that is trained using unsupervised learning and a method to do dimensionality reduction) by segmenting the user’s purchases into T-2, T-1, and T clusters, where T represents the current purchase and T-1 and T-2 represents two previous purchases. For each group of customers, sequential rules are extracted from the purchase sequences of that group using association rule mining to make recommendations. Meanwhile, the segmentation-based KNN-CF method provides recommendations based on the target customer’s purchase data for the current period by selecting the Top-N neighbors from the cluster to which a target user belongs, using binary choice analysis and derive the prediction score of the item not purchased based on the frequency count of the item scanning the purchase data of k-neighbors. Then, the results of the two methods are combined to make final recommendations.

Example of LiuRec09

Let us consider E-commerce historical purchase data containing information of purchase items, frequency of purchase, price and transaction time as input.

Segmentation-based Sequential Rule (SSR) method

Step 1: Customer clustering: Cluster the customers into distinct groups based on their RFM values (Recency, Frequency, and Monetary). Once RFM value for each customer is calculated, all the values are then normalized and K-means clustering method is used to segment all customers based on their normalized RFM values. The RFM patterns of each cluster are identified by assigning "↑" or "↓"; according to whether the RFM value of a cluster is larger than or smaller than the overall average RFM value. An example of clustering customers based on RFM values is demonstrated in Table 2.7.

Table 2.7 Clusters generated by K-means clustering based on the normalized RFM values

	No. of Customers	**R (Recency)**	**F (Frequency)**	**M (Monetary)**	**Patterns**		
Cluster 1	104	72.260	19.587	40797.23	R↑	F↑	M↑
Cluster 2	43	119.558	3.791	7342.326	R↑	F↓	M↓
Cluster 3	17	64.294	67.2351	147315.6	R↓	F↑	M↑
Cluster 4	214	56.696	19.832	40279.53	R↓	F↑	M↑
Cluster 5	78	57.192	37.846	74045.92	R↓	F↑	M↑
Cluster 6	367	58.335	9.632	18677.27	R↓	F↓	M↓
Cluster 7	126	92.246	7.286	14853.89	R↑	F↓	M↓
Cluster 8	240	73.892	8.496	16109.99	R↑	F↓	M↓
Average		68.216	14.324	28638.3			

Clusters with the same pattern are combined into one cluster. For example, clusters 3, 4 and 5 in the Table 2.7 have the same pattern, similarly, clusters 2, 7 and 8 can also be merged. Therefore, eight customer clusters can be reduced to four customer segments - loyal, potential, uncertain, and valueless based on their RFM patterns and is shown in Table 2.8.

Table 2.8 Four customer segments derived by combining clusters with similar RFM patterns

Customer Segments	**No. of Customers**	**R (Recency)**	**F (Frequency)**	**M (Monetary)**
Loyal	309	R↓ (57.239)	F↑ (26.987)	M↑ (54691.80)
Potential	104	R↑ (72.260)	F↑ (19.587)	M↑ (40797.23)
Uncertain	367	R↓ (58.335)	F↓ (9.632)	M↓ (18677.26)
Valueless	409	R↑ (84.347)	F↓ (7.628)	M↓ (14801.23)

Step 2: Transaction clustering: Transactions are divided into groups (transaction clusters) based on similar product items and buying patterns.

- <u>*Transaction matrix creation*</u>

  Once RFM clusters of users are created, user's transaction (binary) matrix is created by analyzing the list of items purchased by users, where 1 represents purchased items and 0 represents non purchased items by a user. An example of transaction matrix created from historical E-commerce data is present in Table 2.9. In this Table 2.9, products are displayed as P1 to P8.

Table 2.9 Transactions recorded by the bit matrix.

Customer	Date	P1	P2	P3	P4	P5	P6	P7	P8	Cluster
C1	20031127	0	0	1	0	0	0	0	0	A
C1	20031127	0	1	0	0	0	0	1	1	B
C2	20040202	0	0	0	1	1	1	0	0	E
C2	20040209	0	1	0	0	0	0	1	0	D
C3	20040126	1	0	0	0	0	0	0	0	C

- *Transaction matrix clustering*

The transaction cluster represents a group of transactions with a similar item purchased by users. First, for each customer make a bit vector. For example, if $user_1$ buys $product_1$ and $product_3$ but did not buy $product_2$ and $product_4$ then its bit vector is (1,0,1,0). The original transactions are transformed into a bit matrix for transaction clustering using SOM clustering technique. Customer's transaction clusters (as shown in the last column of Table 2.9) are used to identify the sequence of transaction clusters over time. A sample change of customers transactions in three periods are displayed in Table 2.10.

Table 2.10 Change in the buying behavior of customer transactions in multiple periods

	Period 1	Period 2	Period 3
Customer 1		AB	E
Customer 2	B		D
Customer 3		A	E

Step 3: Mining customer behavior from transaction clusters: To mine customer behavior according to purchase time, association rule R_j is adopted for determining the most frequent pattern with confidence.

Equation 2.6: Association rule to mine customer behavior in LiuRec09

$$R_j = r_{j,\,T-l+1}, \ldots\ldots\ldots\, r_{j,\,T-1} \rightarrow r_{j,T}\,(Support_j, Confidence_j)$$

where rule R_j indicates that, if the locus of a customer is $r_{j,\,T-l+1}, \ldots\ldots\ldots\, r_{j,\,T-1}$, then the behavior cluster for that customer is $r_{j,T}$ at time T.

From Table 2.10, we can extract a sequential rule $A_{p2} \rightarrow E_{p3}$ (0.4,1) with support of 40 percent and confidence of 100 percent. According to this rule, if a customer's purchase behavior in period P2 is in transaction cluster A, then his/her behavior in P3 will be in transaction cluster E. The other sequential rules $B_{p2} \rightarrow E_{p3}$ (0.2,1) and $B_{p1} \rightarrow D_{p3}$ (0.2,1) can be obtained similarly.

Step 4: The determination and match of the cluster sequences of target customers: The degree of match between a target customer's buying behavior and a sequential rule is calculated by the similarity measure which is, if the behavior of a target customer i is equal to the conditional part of association rule j in the same period, then value is equal to one, otherwise it is equal to zero. Next, this similarity measure is multiplied by the support and confidence of the rule to derive the fitness measure using Equation 2.7.

Equation 2.7: Fitness measure to match target user purchase in LiuRec09

$$SM_y^x = \left(\sum_{k=1}^{l-1} M_{y,T-k}^x\right) \; where \; M_{y,T-k}^x = \begin{cases} 1 \; if \; C_{y,T-k} = r_{x,T-k} \\ 0 \quad otherwise \end{cases} * support_x * \; confidence_y$$

Step 5: Recommendation: Finally, the frequency count of each item in predicted transaction cluster (i.e., the number of transactions in the predicted transaction cluster that contains the product item) is calculated and the top N items with highest frequency count are returned.

Segmentation-based KNN-CF method (SKCF)

In this step for each customer, Pearson's correlation coefficient is used to measure the similarity between target customer and other customers in a same segment and the k most similar (highest ranked) customers are selected as the k-nearest neighbors of the target customer. Then, the N most frequent products not yet purchased by the target customer u (in period T) are selected as the Top-N recommendations.

Hybrid recommendation method

SSR and SKCF are combined linearly with a weighted combination, as shown in Equation 2.8, where α and (1-α) are the weights of SKCF and SSR methods respectively.

Equation 2.8: Weighted combination of SSR & SKCF methods in LiuRec09

$$Product \; Rating = (1 - \; \alpha) * Sequential \; Rule + \; \alpha * Collaborative \; Filtering$$

The product items with the Top-N values of the resulting linear combination of the two methods are selected for recommendation.

2.4.5 Implicit rating-based collaborative filtering and sequential pattern analysis for E-commerce recommendation by Choi, Keunho, Yoo, Kim, & Suh, 2012 (ChoiRec12)

ChoiRec12 proposed a **hybrid recommendation system that uses a combination of CF and SPM**. This system extracts implicit ratings based on purchase history by using the number of times user u purchased item i with respect to total transactions, which can be used in CF even when the explicit rating is not available. CF-based predicted preference (CFPP) of each target user u on an item i is computed as output from the CF process. To make a better recommendation, it also derives sequential patterns from the historical purchase database from which it obtains the output matrix of sequential pattern analysis predicted preference (SPAPP) of each user for each item. The final predicted preference (FPP) of each user for each item is obtained by integrating these two matrices by giving 90% to SPAPP and 10% to CFPP so it can recommend items with highest ratings to users.

Example of ChoiRec12: Let us consider the fragment of historical purchase data as given in Table 2.11, where only purchase time is provided as available information, and our main goal of recommendation is to recommend the suitable item to user T.

Table 2.11 Choi, Yoo, Kim & Suh, 2012 historical user-item matrix

	Item 1	**Item 2**	**Item 3**	**Item 4**	**Item 5**
	Date	**Date**	**Date**	**Date**	**Date**
User 1	01/01	-	01/02	01/03	-
User 2	01/01	-	01/02	01/03	01/04
User 3	-	01/01	01/02	-	01/03
User 4	01/01	01/02	01/03	-	-
User T	-	01/01	01/02	01/03	-

Step 1: Deriving implicit ratings from user transactions: The implicit rating can be computed by: $Implicit\ Rating(u,i) = Round\ up(5 * RP(u,i))$ where, RP(u,i) is the relative preference of user u on item i and it is defined as:

Equation 2.9: Equation for computing relative preference of user u on item i

$$RP\ (u,i) = \frac{AP(u,i)}{\max_{c \varepsilon U}(AP\ (c,i))}$$

where AP(u,i) is the absolute preference of user u on item i and it is defined as:

Equation 2.10: Equation for computing absolute preference of user u on item i

$$AP(u,i) = \frac{number\ of\ transaction\ of\ item\ i\ by\ user\ u}{total\ transactions\ of\ user\ u} + 1$$

In our case, user 1 purchased item 1 one time out of three transactions. Thus, $AP\ (user_1, item_1) = 1/3 + 1 = 1.3$. Furthermore, $RP(user_1, item_1) = \frac{1.3}{1.3} = 1$. So, implicit rating: $RP * 5 = 5$. In the same way, let us consider a user-item implicit rating matrix created from the historical data using above technique as given in Table 2.12.

Table 2.12 Implicit rating derived from user's transactions

	Item 1	**Item 2**	**Item 3**	**Item 4**	**Item 5**	**Mean Rating**
User 1	3	?	1	5	?	3
User 2	4	?	3	1	2	2.5
User 3	?	1	2	?	4	2.3
User 4	5	4	3	?	?	4
User T	?	4	3	2	?	3

Step 2: Calculating mean rating and user similarity based on the implicit rating

- The mean rating is computed by adding all the rating of users on items with respect to total numbers of ratings. So, Mean of rating $User\ 1 = \frac{3+1+5}{3} = 3, User\ 2 = 2.5, User\ 3 = 2.3, User\ 4 = 4$ and $User\ T = 3$.
- Compute similarities between user's using Cosine similarity, which is given as:

Equation 2.11: To compute Cosine similarity

$$Cosine(T, b) = \frac{\sum_{i=1}^{m}(R_{T,i})(R_{b,i})}{\sqrt{\sum_{i=1}^{m}(R_{T,i})^2}\sqrt{\sum_{i=1}^{m}(R_{b,i})^2}}$$

Where ($R_{T,i}$) denote the ratings of users T on item i similarly ($R_{b,}$i) denotes the rating of user b on item i. For example, similarity between target user T and every other user (User 1, User 2, User 3 and User 4) is calculated by using Eq. 2.11. The calculated similarities will be: $CS(T, 1) = 0.7071$ $CS(T, 2) = 0.9648, CS(T, 3) = 0.8944, CS(T, 4) = 1$ where $CS(T, 1)$ means Cosine Similarity between target user T and user 1 and so on.

Step 3: Finding Top k nearest neighbors of target user T: After calculating similarities between target user T and other users, next step is to find top k users as neighbors of T. This is done by sorting the user's similarities in descending order and then selecting the top k (where k=2) neighbors. So, the sorted similarities in descending order will be $CS(T, 4) = 1, CS(T, 2) = 0.9648, CS(T, 3) = 0.8944, CS(T, 1) = 0.7071$. The top 2 neighbors for target user T will be User 4 and User 2.

Step 4: Calculating the CF-based predicted preference (CFPP): The rating information of the top k neighbors is then used to predict CF-based predicted preference of user a on item$_i$ i.e. $CFPP\ (a, i)$, by using Eq. 2.12 where k denotes the number of user a's neighbors and sim (a, b) denotes the similarity between users a and b, and finally, $\overline{R_a}$ and $\overline{R_b}$ represents the mean rating of User a and mean rating of User b. For example, the CFPP of a target user T on all other items will now be CFPP(T, $item_1$) = 4.7455, CFPP(T, $item_2$) = 3.5, CFPP(T, $item_3$) =3.2365, CFPP(T, $item_4$) = 2 and CFPP(T,5) = 3.

Equation 2.12: CF-based predicted preference of user a on $item_i$ in ChoiRec12

$$CFPP\ (a, i) = \overline{R_a} + \frac{1}{\sum_{b=1}^{k} |sim\ (a, b)|} \times \sum_{b=1}^{k} sim\ (a, b) \times (R_{b,i} - \overline{R_b})$$

Step 5: Deriving sequential patterns & computing purchase item-based score (SPAPP)

- Next step is to generate sequence data of each user to calculate predicted preferences (SPAPP) of items. This is done by sorting transaction data for the person according to the transaction date. From Table 2.12, the sequence data for all users except the target user T are: $User_1$: (Item1) (Item3) (Item4); $User_2$: (Item1) (Item3) (Item4) (Item5); $User_3$: (Item2) (Item3) (Item5) & $User_4$: (Item1) (Item2) (Item3).

- Find frequent single item pattern (L_1): Let us consider minimum support as 0.5 then the frequent purchase item (L_1) are $\{< item1 >: 0.75, < item2 >: 0.5, < item3 >: 1, < item4 >: 0.5, < item5 >: 0.5\}$.

- Generate larger candidate set (C_2): Use L_1 Apriori join L_1 to create larger candidates set (C_2) as present in Table 2.13.

Table 2.13 possible list of 2-items generated from frequent purchase (L_1)

Items	**Count**
<item1><item2>	0.25
<item1><item3>	0.75
<item1><item4>	0.5
<item1><item5>	0.25
<item2><item3>	0.5
<item2><item5>	0.25
<item3><item4>	0.5
<item3><item5>	0.5

- Find 2-frequent items from C_2: Test candidate set (C_2) with minimum threshold to create frequent L_2 items. In our case, frequent items are as shown in Table 2.14 and repeat the process of candidate generation (C_k) and pruning (L_k) until the candidate set is empty.

Table 2.14 Frequent 2-item generated from candidate set (C2)

Items	Count
<item1><item3>	0.75
<item1><item4>	0.5
<item2><item3>	0.5
<item3><item4>	0.5
<item3><item5>	0.5

- Match subsequences of a target user purchase with derived purchased items by enumerating target user purchase item. Finally, calculate the pattern analysis based predicted preference (SPAPP) of user T on item i by using the following Eq. 2.13.

Equation 2.13: CF-based predicted preference of user a on item$_i$ in ChoiRec12

$$SPAPP(T, i) = \sum_{S \in SUB} Support_s^i$$

where SUB denotes the set of all subsequences of user T, and $Support_s{}^i$ denotes the support of item i from a subsequence s. For example, SPAPP of target user on item 1 is $SPAPP\ (T, 1)\ =\ 0$, similarly, $SPAPP\ (T, 2)\ =\ 0$, $SPAPP\ (T, 3)\ =\ 0.75 + 0.5 + 0.5 = 1.25$, $SPAPP\ (T, 4)\ =\ 0.5 + 0.5 + 0.5 = 1.5$, $SPAPP\ (T, 5)\ =\ 0.5$.

Step 6: Integrate CFPP and SPAPP: CFPP and SPAPP are normalized to get N_CFPP and N_SPAPP, respectively. Target user T's final predicted preference on $item\ i$, $FPP\ (T, i)$, is calculated using the following Eq. 2.14.

Equation 2.14: Final predicted preference of a user on item in ChoiRec12

$$FPP(T,i) = \alpha * N\_CFPP(T,i) + (1 - \alpha) * N\_SOAPP(T,i)$$

where α and 1- α are weights given to CF and SPA and are set to 0.1 and 0.9 respectively. The FPP values are as shown in Table 2.15.

Table 2.15 Integrating CFPP and SPAPP

	CFPP	SPAPP	N_CFPP	N_SPAPP	FPP
Item 1	4.7455	0.7071	1	0	0.5
Item 2	3.5	0.9648	0.5463	0	0.273
Item 3	3.2365	0.8944	0.4504	0.8333	0.6419
Item 4	2	1	0	1	0.5
Item 5	3	0.333	0.3642	0.3333	0.3488

Step 7: Recommend the item having highest rank: After obtaining FPP values of items purchased by neighbors of the target user, the item having highest FPP is recommended to target user T. In our case

from Table 2.15, if we want to recommend two items, then $item_3$ and $item_4$ will be recommended because they have the highest FPP values.

2.4.6 E-commerce recommendation system based on a hybrid of SPM (prefix span algorithm) & CF (traditional matrix factorization) by Fang, Zhang & Chen, 2016 (Hybrid Model RecSys16)

A **hybrid recommender system that combines the prefix span algorithm with traditional matrix factorization** was proposed. SPM aims to find frequent sequential patterns in sequence database and is applied in this hybrid model to predict customer's payment behavior thus contributing to the accuracy of the model. The workflow of the system consists of three phases: Behavior Prediction Phase, CF Phase and Recommend Phase.

Purchasing Pattern's Extraction

BPM (Behavior pattern model) utilize the Prefix-span algorithm to extract the most prevailing purchasing sequences from the warehouse in real time and match the sequences with the customer's behavior pattern who is browsing or adding an item to cart. Prefix-Span is a pattern growth-based approach, which supports pattern growth by dividing the search space and focusing on the subspace, which requires less memory space for searching. The real time BPM will return a set of the potential purchasing behavior and the category of the purchasing item. When the recommender system's behavior monitoring part detects the user's potential purchasing tendencies, the system will fetch the user's historical behavior record from sequence database and build an item-user rating matrix and each entry contains the historical behavior of the I^{th} user to J^{th} product.

Table 2.16 Item-user rating matrix

Item_Id/User_Id	100011562	100024529	100086267	100637858	100854241
100019569		2	4		1
100022999	1	1	2		
10000003		1			
100009489	3		2	1	3
100018271		1			
100020308			3		

Matrix Factorization-based Collaborative Filtering

CF method is used to find a set of customers whose purchased and rated items overlap the user's purchased and rated items. The algorithm generates recommendations based on a few customers who are most similar to the user and generates the preference tendencies of the users based on their historical purchasing record. The basic matrix factorization model is used which maps both users and items to a joint latent factor space, such that user-item interactions are modeled as inner products in that space. The next step is to factorize this matrix into two matrices, one represents features of the products and another represents the preferences of our users. Multiplying the two matrices, gives back the predictions about user's preference to all products.

Equation 2.15: Equation representing matrix factorization

$$r_{ui} = q_i^T * p_u$$

The r_{ui} represents user u's rating of item i, and the challenge in matrix factorization model is computing the mapping of each item and user to factor vectors q, p, R^F. Since the sparseness of the user-item matrix, SVD is not an appropriate method to decompose the target matrix. Hence, latent factor models (Koren, Bell & Volinsky, 2009) is used to learn the factor vectors (p^u and q^i), by minimizing the regularized squared error on the set of known ratings:

Equation 2.16: Equation for minimizing regularized squared error

$$\min_{p,q} \sum\nolimits_{(u,i)\in \mathrm{k}} (r_{ui} - q_i^T p_u)^2 + \lambda\,(||q_i||^2 \ + ||\,p_u||^2)$$

Recommendation Phase

The payment behavior patterns extracted from the behavior prediction phase and the preference collected from CF method are combined to select target items as suggestions. In the first step, the customer's real-time behavior sequences are generated and stored in database called as candidate database. The candidate database will be scanned at a regular interval and sequence contains payment patterns will be sent to recommender system as potential purchasing sequence. Secondly, for those potential buyers, we will generate the preference information from CF phase which represents the preference degree towards each product. Since sequential mining phase will not only generate the payment sequence, but also the category of the target item, the category matched items in preference vector to recommend will be chosen.

2.4.7 Product recommendation system combining sequential pattern analysis & CF by Jamali & Navaei, 2016 (Product RecSys16)

Proposed a two-level product **hybrid recommendation system which combines C-Means clustering algorithm & Freespan algorithm**. At first, the available products are clustered by using C-Means algorithm to create groups of products with similar characteristics. Then, the second level considers the customer's behavior and their purchase history for drawing the relationships between products by using Sequential Pattern Analysis (SPA) method. These relationships, eventually, will lead to appropriate recommendation for customers and also increases the likelihood of selling related products in electronic transactions.

Proposed PRS (Product Recommendation System) includes two levels of product recommendation: first level is recommended before product purchase and the other one, after purchasing. PRS initially collects product's data from electronic store, separate the products according to their type and are then clustered based on their numerical attributes in three separate clusters of high, medium and low quality by C-means algorithm. Clustering technique is employed to create group of objects based on their features in such a way that the objects belonging to same groups are similar and those belonging in different groups are dissimilar. Here, C-Means clustering algorithm is used to separate products by these types and create groups with similar features and thereby classify products. The algorithm generates clusters based on fuzzy logic and doesn't consider sharp boundaries between the clusters, thus allowing each feature vector to belong to different clusters by a certain degree. The degree of membership of a feature vector to a cluster is usually considered as a function of its distance from the cluster centroid points. It is based on minimization of the following objective function:

Equation 2.17: Equation for minimizing objective function to calculate the degree of membership of a feature vector

$$J_m = \sum_{i-1}^{N} \sum_{i-1}^{C} u_{ij}^{m} \left\| x_i - c_j \right\|^2, 1 \leq m \leq \infty$$

Where m is any real number greater than 1, u_{ij} is the degree of membership of x_i in the cluster j, x_i is the ith of d-dimensional measured data, c_j is the d-dimension center of the cluster, and ||*|| is any norm expressing the similarity between any measured data and the center. Next, the PRS tries to identify customer's requirements and criteria using an online form that takes information about product such as type, quality, price, brand, etc. Thus, this information is used to assign an appropriate cluster to the customer.

In the second level, information about history of customer's shopping behavior is collected. This information is used to explore relations between products by Freespan algorithm of SPA method. Freespan mines sequential patterns by partitioning the search space and projecting the sequence sub-databases recursively based on the projected itemsets (Wei, Jianyong & Han, 2014). Eventually, these relations and patterns will be provided as product recommendations, as it recommends associated products to the products purchased, since the relationships between the products will increase the likelihood of buying the products together and this makes the customer aware of potentially related products.

2.4.8 A sequential pattern-based recommender for product recommendation by Saini, Saumya & Singh, 2017 (SainiRec17)

This work tried to find the sequence of all items which were brought regularly that is not only finding the same product purchased every month, but, also the different products purchased one after another in a sequence. As users buy some products in a sequence, for example, most of the users buy a mobile phone and mobile cover in a sequence. So, the authors tried to find out such kind of sequences, in online shopping. Thus, the main objective of this article is to find out the sequences frequent among all users and Intra-duration in the sequence in an online product purchasing system. With the help of SPADE algorithm, the frequent sequential purchase patterns were found and in the next step, sequence mining algorithm was applied to find out the sequences available in the dataset. Finally, the time elapsed between the purchase of first product and next sequential product was calculated by finding mean and mode of the duration followed by all users. Here, mean gives the average time gap between products, whereas, mode gives the duration followed by most of the users.

2.4.9 E-Commerce product recommendation using historical purchases and clickstream data by Xiao & Ezeife, 2018 (HPCRec18)

A novel recommendation system called Historical Purchase with Clickstream recommendation system (HPCRec) was proposed which integrates purchase frequencies and the consequential bond relationship between clicks and purchases. The term consequential bond was introduced in this HPCRec system and is originated from the concept that customer who clicks on some items will ultimately purchase an item from a list of clicks in most of the cases. By processing this information, it enhances the user-item rating matrix in both quantity and quality aspects and then improves recommendations. The quality of ratings was improved by capturing the level of interest in a product already purchased by a user before, through record of normalized frequency of purchase using the unit vector method. The quantity of ratings was improved with the consequential bond between clicks and purchases, for the sessions without purchases. Finally, the ratings for all the original unknowns are predicted based on this enriched rating matrix using CF algorithm. HPCRec system is capable of providing recommendations for infrequent users and it proves that the consequential bond with the normalized frequencies are more effective at predicting user interest.

Algorithm: Input to HPCRec system are 1) Consequential table (Table 2.17) which shows the relationship between user clicks and purchases and 2) User item purchase frequency matrix (Table 2.19) which represents the frequency of a product purchased from user item rating matrix (Table 2.18). The algorithm is demonstrated below:

Table 2.18 User-item rating matrix

Customer/Item	1	2	3	4
1	?	1	1	?
2	1	1	?	1
3	1	?	?	?

Table 2.17 Consequential table

Session Id	User Id	Clicks	Purchase
1	1	1, 2	2
2	1	3, 5, 2, 3	2, 3
3	2	2, 1, 4	1, 2, 4
4	2	4, 4, 1, 2	2, 4, 4
5	3	1, 2, 1	1
6	3	3, 5, 2	

Step 1: Normalize purchase frequency matrix using unit vector formula: Form user-item purchase frequency matrix (Table 2.19) from Table 2.17, where value represents the number of times product purchased by a user. Normalize purchase frequency to a scaled value (0 to 1) to form Normalized user-item purchase frequency matrix (Table 2.20) using unit vector formula below:

Equation 2.18: Unit vector formula to normalize purchase frequency

$$Frequency\ normalization\ of\ user_u\ on\ item_i = \frac{item\ i}{\sqrt{item_1^2+item_2^2+item_3^2+\cdots+item_n^2}}$$

For example, if user 2 purchases are {item1: 1, item2: 2, item3: 0, item4: 3}, then normalized purchase frequency for user 2 on item 2 is $2/\sqrt{1^2+2^2+0^2+3^2}$=0.53.

Table 2.19 U-I purchase frequency

Customer/Item	1	2	3	4
1	?	2	1	?
2	1	2	?	3
3	1	?	?	?

Table 2.20 Normalized U-I purchase freq matrix

Customer/Item	1	2	3	4
1	?	0.89	0.45	?
2	0.27	0.53	?	0.8
3	1	?	?	?

Step 2: Compute clickstream sequence similarity measurement (CSSM): For each session without a purchase in consequential table, compute clickstream sequence similarity measurement (CSSM) to find similar sessions with purchase value using longest common subsequence rate (LCSR).

Equation 2.19: Longest common subsequence rate

$$LCSR(x,y) = \frac{LCS(x,y)}{max\ (|x|,|y|)}$$

LCS(x,y) is longest common subsequence between $sequence_x$ and $sequence_y$ and is computed by:

Equation 2.20: Longest common subsequence

$$LCS(X_iY_j) = \begin{cases} \emptyset & if\ i = 0\ or\ j = 0 \\ LCS(X_{i-1}Y_{j-1}) \cap x_i & if\ x_i = y_j \\ longest(LCS(X_iY_{j-1}), LCS(X_{i-1}Y_j) & if\ x_i \neq y_j \end{cases}$$

max(|*x*|,|*y*|*)* is maximum length of two sequence.

For example,

$$\mathrm{LCSR}(<3{,}5{,}2>, <3{,}5{,}2{,}3>) = \frac{\mathrm{LCS}(<3{,}5{,}2>, <3{,}5{,}2{,}3>)}{\max(3{,}4)} = \frac{3}{4} = 0.75$$

As there is no purchase information of session 6 in consequential table (Table 2.17), compute Clickstream similarity between session 6 which is <3,5,2> and other sessions & is as shown below:

Table 2.21 CSSM Info table

Clickstream	**Purchase**	**CSSM**
1, 2	2	0.37
3, 5, 2, 3	2, 3	0.845
2, 1, 4	1, 2, 4	0.33
4, 4, 1, 2	2, 4, 4	0.245
1, 2, 1	1	0.295

Step 3: Form a weighted transaction table using the similarity as weight and purchases as transaction records.

Table 2.22 Weighted transactional table of purchase set created from consequential bond

Purchase	<2>	<2, 3>	<1, 2, 4>	<2, 4, 4>	<1>
1	0.37	0.845	0.33	0.245	0.295

Step 4: Call TWFI (Transaction-based Weighted Frequent Item) function, which takes a weighted transaction table, where weights are assigned to each transaction as input and returns items with weighted support in a given threshold. For example, let's consider minimum weighted support=0.1, then, we will have frequent weighted transaction table as shown in Table 2.23.

Table 2.23 Weighted frequent transactional table

Purchase (Transaction Records)	2	2, 3	1, 2, 4	2, 4, 4	1
Weight	0.37	0.845	0.33	0.245	0.295

Step 5: Calculate support to form a distinct item from set of all the transactions

Table 2.24 Support for item present in weighted frequent transaction table

Item	1	2	3	4
Support	2	4	1	3

Step 6: Compute the average weighted support for each item using (AWS=AW*support), where $AW = sum(weight)/support$). For example, AWS (1) =0.33+ 0.295=0.625, AWS (4) =0.33+ 0.245+0.245=0.82.

Table 2.25 Weight for item present in purchase pattern

Item	1	2	3	4
AWS	0.625	1.79	0.845	0.82

Step 7: Normalize weighted support using feature scaling

Equation 2.21: Equation for feature scaling

$$x' = \frac{\mathrm{x} - \mathrm{min}}{\mathrm{max} - min}$$

So for the average weighted support, max=1.79, min=0.625, then the new average weighted support for item_3 is (0.845−0.625)/(1.79−0.625) = 0.189, and all the weighted supports are <1 : 0, 2 : 1, 3 : 0.189, 4 : 0.167>

Step 8: Return all the items that have a normalized weighted support greater than or equal to minimum weighted support (e.g., (2:1),(3:0.189),(4:0.167)). Then for each one of these items, if user has not purchased it, add the weight into the normalized user-item matrix.

Step 9: Return to step 2 if there are more sessions without a purchase, otherwise, run the CF algorithm using the updated rating matrix to get predicted ratings for all of the original unknowns as demonstrated in Table 2.26.

Table 2.26 User-item rating matrix with predicted ratings

	Item 1	Item 2	Item 3	Item 4
User 1	0.63	0.89	0.45	0.49
User 2	0.27	0.53	0.35	0.8
User 3	1	0.74	0.27	0.33

2.4.10 E-Commerce product recommendation using historical sequential patterns and clickstream data by Bhatta, Ezeife & Butt, 2019 (HSPRec19)

This work was proposed to improve the HPCRec system which did not integrate frequent sequential patterns to capture more real-life customer sequence patterns of purchase behavior inside consequential bond. Thus, the authors proposed an algorithm called HSPRec (Historical Sequential Pattern Recommendation System), which explored enriching the user-item matrix with sequential pattern of customer clicks and purchases to capture better customer behavior. HSPRec takes minimum support, historical user-item purchase frequency matrix and consequential bond as input and the sequential database of purchases and clicks was mined with the GSP algorithm to discover frequent historical sequential patterns to improve consequential bond between clicks and purchases and enhance user-item frequency matrix quantitatively and qualitatively to generate a rich user-item matrix for CF to further improve recommendations.

Example of HSPRec

Table 2.27 Consequential table from click and purchase historical data

User Id	Click	Purchase
1	Cheese, Butter, Milk, Butter, Cream, Cheese, Honey, Cream, Butter	Cream, Butter, Milk, Honey, Butter
2	Cheese, Cream, Honey, Butter	Butter, Cheese, Cheese, Honey
3	Cheese, Milk	?

Let us consider the consequential bond of clicks and purchases (Table 2.27) created from click and purchase historical data and daily sequential database (Table 2.28) created from historical transaction data by considering the period of time (day, week, and month).

Table 2.28 Daily sequential database created from click and purchase historical data by considering the period of time

SID	Click Sequence	Purchase Sequence
1	<(Cheese, Butter, Milk, Butter, Cream, Cheese), (Honey, Cream, Butter)>	< (Cream, Butter, Milk), (Honey, Butter)>
2	<(Cheese, Cream, Honey, Butter)>	<(Butter, Cheese), (Cheese, Honey)>
3	<(Cheese, Milk)>	?

Algorithm:

Step 1: Create a user-item purchase frequency matrix (Table 2.29) from Table 2.27, where the number indicates, the number of times item purchased by a user. For example, User 1 purchased butter twice, Honey once and so on.

Table 2.29 User-item frequency matrix created from Table 2.27

User/item	Milk	Bread	Butter	Cream	Cheese	Honey
User 1	1	?	2	1	?	1
User 2	?	?	1	?	2	1
User 3	?	?	?	?	?	?

Step 2: Create frequent sequential purchase patterns from daily sequential database (Table 2.28) using GSP algorithm. In this case, the possible purchase sequential rules from frequent purchase sequences are

Table 2.30 Sequential rules created from n-frequent sequences

Rule No	Sequential rule
1	Milk, Butter → Cheese
2	Cream, Cheese → Milk
3	Cheese, Honey → Cream
4	Honey → Cream
5	Honey → Milk

Step 3: Fill purchase information in user-item frequency matrix using sequential purchase rules.

Table 2.31 Rich user-item frequency matrix created with help of sequential rule

User/item	Milk	Bread	Butter	Cream	Cheese	Honey
User 1	1	?	2	1	**1**	1
User 2	**1**	?	1	**1**	2	1
User 3	?	?	?	?	?	?

Step 4: As it can be seen in Table 2.28 that there is no purchase information of user 3, to find purchase information of user 3, analyze the relationship between click and purchase considering their sequence using the following steps:

1. Use an SPM algorithm on user click sequence: Create n-frequent click sequential patterns from click sequences of Table 2.28 using the GSP algorithm. In this case some of the n-frequent click sequences are:

   1- Sequences = {< (Milk)>, < (Cheese)>, < (Cream)>, < (Butter)>, < (Honey)>}
   2- Sequences = {< (Milk, Cheese)>, < (Butter, Cheese)>, < (Honey, Butter)>}
   3- Sequences = {< (Cheese, Cream, Milk)>, < (Cream, Cheese, Milk)>}

2. Create sequential rules (Table 2.32) from n-frequent click sequential patterns using Sequential Pattern Rule (SPR) method. In this case, the possible sequential rules from n-frequent click sequences are

Table 2.32 Sequential rules created from n-frequent sequences

Rule No	Sequential rule
1	Cheese, Milk → Cream
2	Cream → Cheese
3	Butter → Honey

3. Recommend item from the click sequential rule, where the user clicks but does not purchase anything. For example, there is no purchase for click sequence < (Cheese, Milk)> thus item < (Cream)> is recommended from the sequential rule (Rule no:1 from Table 2.32)

Step 5: Compute Click Purchase Pattern (CPS) similarity using frequency and sequence of click and purchase patterns. If there is no purchase along with click item, then use the recommended item. For example, let's take click (X) = **{<(Cheese, Butter, Milk, Butter, Cream, Cheese)>, < (Honey, Cream, Butter)>}** by user 1 and purchase (Y) = {<**(Cream, Butter, Milk), (Honey, Butter)**>}.

i. Calculate $LCSR(X,Y) = \frac{common(X,Y)}{max(X,Y)} = \frac{5}{9} = 0.55$

ii. Calculate $FS(X,Y) = Cosine(\{2,1,1,1\},\{1,0,2,2,1,3\}) = \frac{10}{10.21} = 0.97$; where X= {Milk :1, Bread :0, Cream :2, Cheese :2, Honey :1, Butter :3 } and Y={ Milk :1, Bread :0, Cream :1, Cheese :0, Honey :1, Butter :2 } are frequency of products present in X and Y

iii. Use α and β as parameters to balance the sub sequence similarity and frequency similarity, where $0 < \alpha, \beta < 1, \alpha + \beta = 1$. α and β will be determined from training dataset. So, if set $\alpha = 0.8, \beta = 0.2, CPS - Sim\ (X,Y) = 0.8 * 0.55 + 0.2 * 0.97 = 0.634$ (Table 2.33)

Table 2.33 CPS similarity using click and purchase

User Id	Click	Purchase	Recommend Item	CPS similarity
1	<(Cheese, Butter, Milk, Butter, Cream, Cheese), (Honey, Cream, Butter)>	<(Cream, Butter, Milk), (Honey, Butter)>		0.634
2	<(Cheese, Cream, Honey, Butter)>	<(Butter, Cheese), (Cheese, Honey)>		0.562
3	<(Butter, Bread, Cream, Cheese, Honey, Butter)>	?	<(Cream)>	0.198

Step 6: Assign Click Purchase (CPS) similarity value to the purchase patterns present in the consequential bond.

Step 7: Assign weighted purchase patterns to Weighted Frequent Purchase Pattern Miner (WFPP) and compute a weight for item present in weighted purchase pattern using formula (eq. 2.21):

Equation 2.21: Formula to compute weight in WFPPM

$$R_{item_i} = \frac{\sum_{i=1}^{n} CPS\ containing\ item_i}{Support\ (item_i)}$$

i. Count support of item:

Table 2.34 Support for item present in weighted purchase patterns

Item	Milk	Cream	Cheese	Honey	Butter
Support	1	1	2	2	3

ii. Calculate rating for individual item:

$$R_{milk} = \frac{0.634}{1} = 0.634$$
$$R_{cream} = \frac{0.634}{1} = 0.634$$
$$R_{cheese} = \frac{0.562 + 0.562}{2} = 0.562$$
$$R_{Honey} = \frac{0.634 + 0.562}{2} = 0.598$$
$$R_{Butter} = \frac{0.634 + 0.634 + 0.562}{3} = 0.61$$

Step 8: Use the weight of item to make user-item matrix rich. In our case, rich user-item purchase frequency matrix is shown in Table 2.35.

Table 2.35 Rich user-item purchase frequency matrix

User/item	Milk	Bread	Butter	Cream	Cheese	Honey
User 1	1	?	2	1	**1**	1
User 2	**1**	?	1	**1**	2	1
User 3	0.63	?	0.61	0.63	0.56	0.59

Step 9: Normalize rich user-item purchase frequency matrix to get normalized quantitatively rich user-item matrix (Table 2.36) using unit normalization function given below in Equation 2.22.

Equation 2.22: Unit normalization function

$$Normalization\ (r_{ui}) = \frac{r_{ui}}{\sqrt{r_{ui_1}^2 + r_{ui_2}^2 + \cdots r_{ui_n}^2}}$$

Table 2.36 Quantitatively rich purchase user-item purchase frequency matrix

User/item	Milk	Bread	Butter	Cream	Cheese	Honey
User 1	0.35	?	0.70	0.35	**0.35**	0.35
User 2	**0.35**	?	0.35	**0.35**	0.70	0.35
User 3	0.48	?	0.53	0.38	0.47	0.40

CHAPTER 3: REVIEW OF SEQUENTIAL PATTERN-BASED E-COMMERCE RECOMMENDATION SYSTEMS

Upon conducting a systematic review of SP-based E-commerce RS to have a comprehensive understanding of the recommendation paradigm, we identified the answers for research questions that we posed in (section 1.6) chapter 1.

(1) How has SPM been used to handle sparsity problem, improve recommendation accuracy, novelty and scalability in the reviewed systems?

In E-commerce recommendation systems, the number of ratings already obtained is usually very less when compared to the number of ratings that needs to be predicted. This results in a sparse user-item matrix and generates weak or poor recommendations as a result of insufficient rating information. Analysing historical sequential purchase patterns of a user using SPM, provides the relationship between already purchased items and recommended items to fill the missing rating for an item to improve the user-item matrix quantitatively (providing possible value for the unrated item or 0 value item in user-item matrix) and qualitatively (indicating the level of user-interest on an item) thus, handling the sparsity problem. By processing frequent clicks and/or purchase sequential patterns generates a rich user-item matrix for CF algorithm to further improve recommendations in terms of accuracy, novelty and scalability.

(2) What are the existing challenges faced by E-commerce recommendation systems and how they can be solved?

Some of the problems associated with CBF techniques are limited content analysis, overspecialization and sparsity of data, CF techniques exhibit cold-start, sparsity and scalability problems and Knowledge-based methods face overspecialization issue. Also, these traditional recommendation systems like CF & CBF techniques cannot capture the changes in purchase behavior of the customers over time, whereas SPM, a knowledge-based method can capture this. As these aforementioned problems usually reduce the quality of recommendations, Hybrid approach has been proposed which combines two or more techniques in different ways in order to mitigate some of the problems identified, by harnessing their strengths and increase the accuracy and performance of recommendation systems with respect to diversity and novelty.

(3) What is the importance of SPM in recommendation systems for the application domain of E-commerce?

User-item interactions in E-commerce domain are essentially sequentially dependent as shopping usually happens successively in a sequence. These sequential dependencies cannot be well captured by conventional recommendation systems like CBF & CF techniques which essentially accentuates the importance of SPM to discover temporal patterns that are frequently repeated among different users, from historical purchase sequences. SPM adds an additional dynamic by taking the order of previous interactions into account. Modeling of these sequential dependencies facilitates a more engaging user experience, resulting in recommendations that are more responsive to recent user and item dynamics.

Taxonomy for existing SP-based E-commerce RS is proposed based on the following three categories, in this chapter.

3.1 Effect of Sequential Patterns on recommendation systems with respect to improving the quality and quantity of user-item rating matrix input

Recommendation systems in E-commerce suffer from uninformative rating data which usually only represents if a user has purchased a product before. This user-item rating matrix is usually sparse, less informative and leads to poor recommendations (Bucklin & Sismeiro, 2003). Thus, in order to capture more real-life customer purchase behavior and to provide the relationship between already purchased items and recommended items, historical sequential purchase patterns of a user are analyzed and integrated into the user-item matrix input to enhance and improve the rating quality (specifying level of interest or value for already rated items) and quantity (finding possible rating for previously unknown ratings) by using mined sequential patterns (discussed in detail in section 1.3 of chapter 1). Table 3.1 shows how the surveyed recommendation systems improved the quality and quantity of user-item rating matrix input with the use of sequential patterns in comparison to each other.

Table 3.1 Summary of how the surveyed recommendation systems improved the quality and quantity of input user-item (U-I) rating matrix

Recommendation System	**Improving rating quality**	**Improving rating quantity**
ChoRec05	No use of historical purchases or clickstream data which mines the user purchase behavior that can be integrated into the U-I	Used association rule mining technique for predicting the possibility of purchase

	rating matrix to improve the rating quality	
ChenRec09	Used RFM - Recency, Frequency and Monetary concept for generating information about customer purchase behavior to improve the rating quality	Used modified Apriori algorithm to extract the sequential patterns from customer’s purchase data for predicting the possibility of purchase
HuangRec09	No use of historical purchases or clickstream data which mines the user purchase behavior that can be integrated into the U-I rating matrix to improve the rating quality	Association rule mining technique was used for predicting the possibility of purchase
LiuRec09	Used RFM - Recency, Frequency and Monetary concept for generating information about customer purchase behavior to improve the rating quality	Used association rule mining technique to derive the sequential rules in order to predict the possibility of purchase
ChoiRec12	Used historical purchases, frequency of the purchase and relative preference of a user u on item I to improve the rating quality	Sequential rules derived from historical purchase database using association rule mining are used for predicting the possibility of purchase
Hybrid Model RecSys16	Consumer's different behaviors like click, collect, add to cart and payment were incorporated to extract the user's potential purchasing tendencies and thereby an item-user rating matrix was built	Sequential patterns derived with the help of Prefix-Span algorithm are used for detecting the user's potential purchase tendencies and predicting the possibility of purchase in U-I rating matrix
Product RecSys16	Used historical purchases and frequency of the purchase to improve the rating quality	Freespan algorithm was used to extract sequential patterns from the purchase history for drawing a relationship between products and thereby predicting the possibility of purchase
SainiRec17	Used historical purchases to improve the rating quality	With the help of SPADE algorithm, frequent sequential purchase patterns were found, and the possibility of purchase was predicted thereafter
HPCRec18	Used user’s historical purchases, frequency of the purchase, clickstream behavior and consequential bond	Analysed the session-based consequential bond of historical clicks and purchases of a user to provide the relationship between already

	information of historical clicks and purchases of a user to improve the rating quality between the values 0 and 1	purchased items and recommended items to fill the missing rating of an item by providing possible value for the unrated item
HSPRec19	Used user's historical sequential purchases, frequency of the purchase, clickstream behavior and consequential bond information of historical clicks and purchases of a user to improve the rating quality between the values 0 and 1	Used GSP algorithm to extract sequential purchase patterns of a customer and mined the consequential information between clicks and purchases to fill the missing rating of an item by providing a possible value (0.5) for unrated item

3.2 Effect of Sequential Patterns on recommendation systems with respect to handling the problems of sparsity, novelty and scalability

In academic environments, the evaluation of recommendation systems performance is dominated by simulation-based experiments on historical rating or implicit feedback datasets. The quality of the output of an algorithm can then be assessed with the help of accuracy metrics. Being able to accurately predict the relevance of items for users is and will be a central problem of recommendation systems research. Increasing the prediction accuracy therefore is a relevant goal of research (Jannach & Jugovac, 2019). But accuracy alone is not enough! Recommending items that the user might have bought anyways might be of little business value. Hence, focusing on accuracy alone can lead to monotone recommendations and limited discovery. Thus, it is important that the recommendation systems can assess multiple, possibly competing goals in parallel such as handling data sparsity, improving novelty and scalability of the recommendation systems.

Sparsity: In practice, many commercial recommendation systems (e.g., book recommendation in Amazon.com) are used to evaluate large product sets. In these systems, even active customers may have purchased only under 1% of the products (1% of 2 million books is 20, 000 books) i.e. only a few of the total number of items available in a database are often rated by users. Thus, in any recommendation system, the number of ratings already obtained is usually very less when compared to the number of ratings that needs to be predicted. This results in a sparse user-item matrix and generates weak or poor recommendations as a result of insufficient rating information.

Novelty: The novelty evaluates the likelihood of a recommendation system to give recommendations to the user that they are not aware of, or that they have not seen in the past. Unseen recommendations often increase the ability of the user to discover important insights into their likes and dislikes that they did not know previously. This is more important than discovering items that they were already aware of but have not rated them.

Scalability: It has become increasingly easy to collect large number of ratings and implicit feedback information from various users in recent years. In such cases, the size of the data set continues to increase over time. As a result, it has become increasingly essential to design recommendation systems that can perform effectively and efficiently in the presence of large amounts of data. The importance of scalability has become particularly great in recent years because of the increasing importance of the "big-data" paradigm.

A taxonomy for SP-based E-commerce RS is developed and is provided in Table 3.2, which shows the effect of SP on surveyed recommendation systems performance by examining how all the surveyed algorithms handled the problems like sparsity, novelty, scalability and improving the User-Item (U-I) rating quality and quantity of recommendation systems with the use of sequential patterns in comparison to each other. The interpretation of the terms high, medium and low (in Table 3.2) with respect to the individual functionalities is defined below, followed by an explanation as to why these systems are in a specified range.

Table 3.2 Effect of SP on surveyed recommendation systems performance with respect to handling the problems of sparsity, novelty, scalability and recommendation input

Recommendation System/ Performance Factors	Reducing Data Sparsity	Improving Novelty	Improving Scalability	Improving U-I rating quality	Improving U-I rating quantity
ChoRec05	Low	High	Medium	Low	Low
ChenRec09	Medium	Low	Low	Medium	Medium
HuangRec09	Low	High	Medium	Low	Low
LiuRec09	Low	High	Medium	Low	Low
ChoiRec12	Medium	Low	Low	Medium	Medium
Hybrid Model RecSys16	Medium	Low	Medium	Medium	Medium
Product RecSys16	Medium	High	High	Medium	Medium
SainiRec17	Medium	Low	Low	Medium	Medium
HPCRec18	High	Low	Low	High	High
HSPRec19	High	High	Medium	High	High

<u>Reducing Data Sparsity</u>

Low: No use of SPM, instead Association rule mining was used

Medium: Used SPM but couldn't integrate any other implicit user behavior like clickstream data etc.

High: Used SPM and integrated additional behavioral data like clickstream data to enhance user/item matrix

<u>Improving Novelty</u>

Low: previously purchased items by the target user were also included in the recommendation list

High: known items were excluded from being recommended and associated products to purchased products were used for recommendation purposes to make the customer aware of potentially related products

<u>Improving Scalability</u>

Low: No clustering technique was used to reduce the dimensionality of the dataset

Medium: A clustering technique was used to reduce the dimensionality of the dataset

High: A clustering technique along with an additional dimensionality reduction technique was used

<u>Improving U-I (User-Item) rating quality & quantity</u>

Low: No user's historical purchases, clickstream behavior, frequency of the purchase or other user purchase behavior was mined to be integrated into the U-I rating matrix

Medium: Minimum information such as only one among the user purchase behavior like association rules (which are not as informative as sequential patterns), user's historical purchases, clickstream behavior, frequency of the purchase are incorporated into the U-I rating matrix which is a less complex method of mining user purchase behavior

High: More informative customer purchase historical behavior features are mined and incorporated into U-I rating matrix such as clickstream behavior, consequential bond information of historical clicks and purchases of a user, historical sequential purchase behavior (sequential patterns) etc.

The early hybrid recommendation systems like ChoRec05, ChenRec09, HuangRec09, LiuRec09 and ChoiRec12 used association rule mining for improving the quality of rating input. None of these systems incorporated additional customer behavioral data like clickstream data or browsing history from which implicit behavior can be extracted and is used to fill the unknown ratings. Hence these systems are assigned a "low" level on reducing data sparsity. HSPRec19 system could achieve this to a higher extent by using SPM (GSP algorithm) to derive sequential patterns for improving the rating quality and quantity. Thus, this system is assigned a “high” level on reducing data sparsity. The remaining four systems (Hybrid Model RecSys16, Product RecSys16, SainiRec17 and HPCRec18) didn’t integrate any additional behavior but extracted the sequential patterns using SPM algorithms like PrefixSpan, FreeSpan and SPADE which resulted in reducing data sparsity to a “medium” level. The novelty rate is defined “low” if the previously purchased items by the target user were included in recommendation list because novelty accounts for the likelihood of a recommendation system to give recommendations to the user that they are not aware of. Thus, the novelty rate is defined “high” if the known items were excluded from being recommended and associated products to the purchased products were used for recommendation purposes to make the customer aware of potentially related products. The dimensionality of a dataset is reduced by using either a clustering technique or by explicitly using a dimensionality reduction technique and sometimes both. Downsizing the data dimension leads to an increase in the scalability of the recommendation system. Thus, if no clustering technique was used by the system, then improving the scalability was specified as “low” and if a clustering technique was used to reduce the dimensionality of the dataset then improving the scalability was specified to be “medium” and if a clustering technique along with an additional dimensionality reduction technique was used by the system then improving the scalability was specified “high”.

Nevertheless, the actual quantification of some of these factors is often quite subjective, and there are often no hard measures to provide a numerical metric. From a quantification perspective, accuracy is a concrete goal that is relatively easy to measure and is therefore used more frequently for benchmarking and testing. The set of metrics commonly used to assess the quality/performance of a recommendation algorithm are discussed below in the next category of the proposed taxonomy.

3.3 Effect of Sequential Patterns on the performance of Recommendation Systems

Prediction accuracy is by far the most discussed property to evaluate the performance of a recommendation system in the literature. A basic assumption is that a system that provides more accurate predictions will be preferred by the user and hence, many researchers set out to find algorithms that provide better predictions in terms of accuracy metrics. The quality of a recommendation system algorithm can be assessed with the help of accuracy metrics/measures such as Mean Absolute Error (MAE), or Precision and Recall which would be discussed below. Accuracy metrics are used to evaluate either the prediction accuracy of estimating the ratings of specific user-item combinations or the accuracy of the top k rankings predicted by a recommendation system. Let R be the ratings matrix in which r_{uj} is the known rating of user u for item j. Consider the case where a recommendation algorithm estimates this rating as $\hat{r}_{uj}$. Then, the entry-specific error of the estimation is given by the quantity $e_{uj} = \hat{r}_{uj} - r_{uj}$. The overall error is computed by averaging the entry-specific errors either in terms of absolute values or in terms of squared values. An example is the *mean absolute error*, which is denoted by *MAE*.

Mean Absolute Error: MAE measures the average of errors in a set of predictions, i.e. it's the average of the absolute differences between prediction and actual rating over the test sample. Thus, higher mean absolute errors mean, less efficient for accurate rating prediction and lower mean absolute errors means highly efficient for accurate rating prediction.

$$MAE = \frac{\sum_{i=1}^{n} |actual\_rating - predicted\_rating|}{n}$$

Now, let us consider the confusion matrix as shown below:

Table 3.3 Confusion Matrix

	Not Purchased	**Purchased**
Not recommended (Not relevant)	**TN** (Not recommended and Not Purchased)	**FP** (Not recommended and Purchased)
Recommended (relevant)	**FN** (Recommended and Not Purchased)	**TP** (Recommended and Purchased)

Precision: Determines the fraction of relevant items retrieved out of all items in an RS. Let us consider, TP represents the fraction of items that user is interested with and FP represents the fraction of items that user is not interested with, then precision is defined as:

$$Precision = \frac{TP}{TP + FP}$$

Recall: Determines the fraction of relevant items retrieved out of all relevant items in an RS. Let us consider, TP represents the fraction of relevant items that a user is interested with and FN represents the fraction of relevant items that a user is not interested with, then recall is defined as:

$$Recall = \frac{TP}{TP + FN}$$

F1 score is the weighted average of precision and recall.

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Table 3.4 provides the summary of performance of the surveyed SP-based E-commerce RS with regards to recommendation accuracy metrics like precision, recall and MAE. Their evaluations were carried out on different datasets.

Table 3.4 Summary of performance of surveyed recommendation systems in terms of recommendation accuracy metrics like precision, recall and MAE

Recommendation System	**Accuracy Metrics**			
	Precision	**Recall**	**F1**	**MAE**
ChoRec05	0.03	0.12	0.04	NA
HuangRec09	0.06	0.05	0.01	NA
LiuRec09	Wasn't evaluated		0.04	NA
ChoiRec12	0.30	0.45	0.23	0.64
Hybrid Model RecSys16	0.15	0.04	0.19	NA
Product RecSys16	Accuracy – 86.91%			NA
HPCRec18	0.37	0.60	0.46	0.52
HSPRec19	0.44	0.75	0.55	0.30

It can be observed that, with the progress in this research field, performance of the recommendation systems was improved gradually in comparison to each other, as early hybrid recommendation systems like ChoRec05, HuangRec09, LiuRec09 and ChoiRec12 extracted some historical purchase sequences to analyze how the customer's purchase behavior may vary over time using the sequential rule-based methods rather than discovering the sequential patterns with the help of SPM algorithms which can capture better customer behavior by including user's sequential purchase or click stream behavior in the user-item interaction. However, the hybrid recommendation systems like Hybrid Model RecSys16, Product RecSys16, HPCRec18 & HSPRec19 used SPM algorithms PrefixSpan, FreeSpan & GSP algorithms to capture more real-life customer purchase behavior by extracting the complex sequential patterns of user purchase behavior and these patterns are used to fill the missing ratings in user-item matrix input so that the input becomes more informative before it is fed to CF and thereby reducing the sparsity level. HSPRec19 system performed the best in comparison to other recommendation systems using the GSP algorithm for mining sequential database of purchases and clicks to discover frequent historical sequential patterns to improve the consequential bond between clicks and purchases in order to capture better customer behavior and enhance user-item frequency matrix quantitatively and qualitatively, generating a rich user-item matrix for CF to further improve recommendations in terms of data sparsity, novelty and scalability of recommendation systems alongside improving the accuracy of recommendations with the use of sequential patterns.

CHAPTER 4: COMPARATIVE STUDY AND ANALYSIS OF SEQUENTIAL PATTERN-BASED E-COMMERCE RECOMMENDATION SYSTEMS

A comparative study of the existing SP-based E-commerce RS is provided in Table 4.1 which provides a discussion of their corresponding working mechanisms and the limitations.

Table 4.1 Comparative study of the surveyed SP-based E-commerce RS

Recommendation System	Recommendation Method	Limitations
ChoRec05 (Cho, Cho & Kim, 2005)	Proposed a hybrid approach that combines CF (SOM clustering) & Sequential cluster rules extraction using Sequential Rule Mining	Does not consider customer segmentation which would have improved the quality of the sequential rule-based (SR) method by making recommendations based on customer groups.
HuangRec09 (Huang et al., 2009)	Proposed a hybrid two-stage recommendation system to predict the customer's time-variant purchase behavior with respect to both the product category as well as the product item that combines CF (GA based clustering approach) & Sequential rule-based method	Identification of the dynamic purchase behaviors of customers that purchase goods infrequently is difficult. Also, this system cannot handle the multiple categories problem via Sequential rule-based method in the stage of product categories prediction.
LiuRec09 (Liu, Lai, & Lee, 2009)	Proposed a hybrid recommendation system that combines segment-based sequential rule mining with segment based KNN CF	Only finds the transaction cluster changes but not all the sequential rules and there is no provision for recommending infrequent items.
ChoiRec12 (Choi, Yoo, Kim & Suh, 2012)	Proposed a hybrid recommendation system that extracts implicit ratings from purchase data so that CF is applied, and the sequential rules are derived from historical purchase database. Results of two methods are combined by giving 90% to SPA and	User purchase sequential patterns are not considered during the user-item matrix creation and there is no provision for recommending infrequent items.

	10% to CF for recommending items with highest ratings to the users	
Hybrid Model RecSys16 (Fang, Zhang & Chen, 2016)	Proposed a hybrid recommendation system that combines the SPM (prefix span algorithm) with CF (traditional matrix factorization) to predict customer's purchasing behavior	Could not obtain personalized information as the recommending model wasn't varied for different groups of users or items.
Product RecSys16 (Jamali & Navaei, 2016)	Proposed a hybrid two-level product recommender which combines CF (C-Means clustering algorithm) & SPM (Freespan algorithm)	Cannot provide recommendations unless multiple item purchasing profiles for a number of consumers, or at least for the consumer currently using the system, are available.
HPCRec18 (Xiao & Ezeife, 2018)	Proposed a Clickstream based CF recommender system to improve the quality of user-item matrix by normalizing the frequency of item purchase. Each session-based click sequences are then matched to a purchase and for those without a purchase, the purchase possibility is derived by analysis of consequential bond. Finally, the ratings are predicted based on this enriched rating matrix	Unable to integrate sequential patterns during qualitative and quantitative analysis of user-item matrix.
HSPRec19 (Bhatta, Ezeife & Butt, 2019)	Proposed a hybrid recommender system which explored enriching the user-item matrix with sequential patterns of customer clicks and purchases using GSP algorithm to capture better customer behavior and the enhanced user-item matrix is then fed to CF for further improving recommendations	Unable to incorporate multiple data source based sequential patterns. Also, there's no provision for infrequent users.

4.1 Comparative analysis of Traditional CF, ChoiRec12, HPCRec18 & HSPRec19 systems with respect to precision, recall and MAE (Bhatta, Ezeife & Butt, 2019)

(Bhatta, Ezeife & Butt, 2019) used user-based collaborative filtering to compare and evaluate the performance of recommendation systems (ChoiRec12, HPCRec18, and HSPRec19). First, the user-based CF was applied on explicit rating available on Amazon historical data which consisted of 23 different categories such as **Books, Electronics, Home and Kitchen, Sports and Outdoors, Cell Phones and Accessories, Grocery and Gourmet Food and many more.** The data contained 142.8 million transactional records spanning from May 1996 - July 2014. The Pearson Correlation Coefficient (PCC) has been used to test user-based CF. The historical data was then converted into user-item matrices with ChoiRec12, HPCRec18, and HSPRec19 algorithms and finally was provided to CF. This modified historical dataset was then used to evaluate the performance of ChoiRec12, HPCRec18, and HSPRec19 recommendation systems with respect to MAE, precision, and recall.

4.1.1 Choosing similarity measure

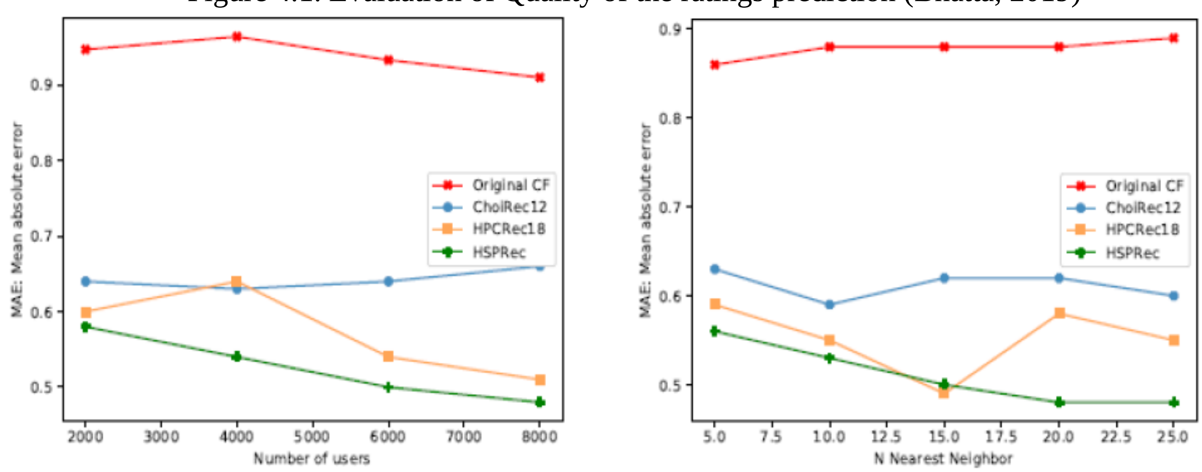
In order to calculate similarity between a target user and every other user, as is done in a traditional CF technique, similarity functions such as Pearson correlation coefficient or cosine similarity or distance measures are used. The choice of similarity function should be made properly based on the data set at hand. The **Pearson correlation coefficient** estimates the similarity based on the rating pattern between two users and is a measure of the strength of a linear association between two variables i.e. it indicates to which extent two variables are linearly related. **Cosine similarity** treats two users as two vectors in the m-dimensional rating vector space, where m denotes the set of all items rated by both users and estimates the similarity by calculating the cosine value of the angle between the two vectors. Finally, **distance measure** estimates the similarity between a target user and other users by calculating the absolute magnitude of the similarity between two users in the m-dimensional rating vector space, so that distance-based similarity is defined as an inverse of the distance. Since the above three similarity functions estimate the similarity between two users from different perspectives, depending on the similarity functions to be used, the set of neighbors whose rating information is used to predict the preference of a target user on candidate items to recommend could be different, and thus, so are the items finally recommended. To find a similarity function that is more appropriate for a data set, I recommend using all the three similarity functions to compare their accuracies and then decide a similarity measure accordingly.

4.1.2 Result evaluation and analysis

Initially, (Bhatta, Ezeife & Butt, 2019) applied the user-based CF on explicit ratings of Amazon historical data and they observed that the performance was very low. Then, the authors implemented ChoiRec12 (Choi, Keunho, Yoo, Kim, & Suh, 2012) with the derived implicit ratings and got a better result compared to traditional CF. Furthermore, HPCRec18 (Xiao, 2018) was implemented and a better result was obtained than ChoiRec12. Finally, HSPRec19 (Bhatta, Ezeife & Butt, 2019) was implemented with the help of purchase frequency matrix at first. Then, frequent sequences of purchase data were discovered to create sequential rules and these sequential rules were used to enhance user-item matrix and then the CF was applied and found better result compared to ChoiRec12 and HPCRec18. But, with the increase in number of users, the performance decreased gradually.

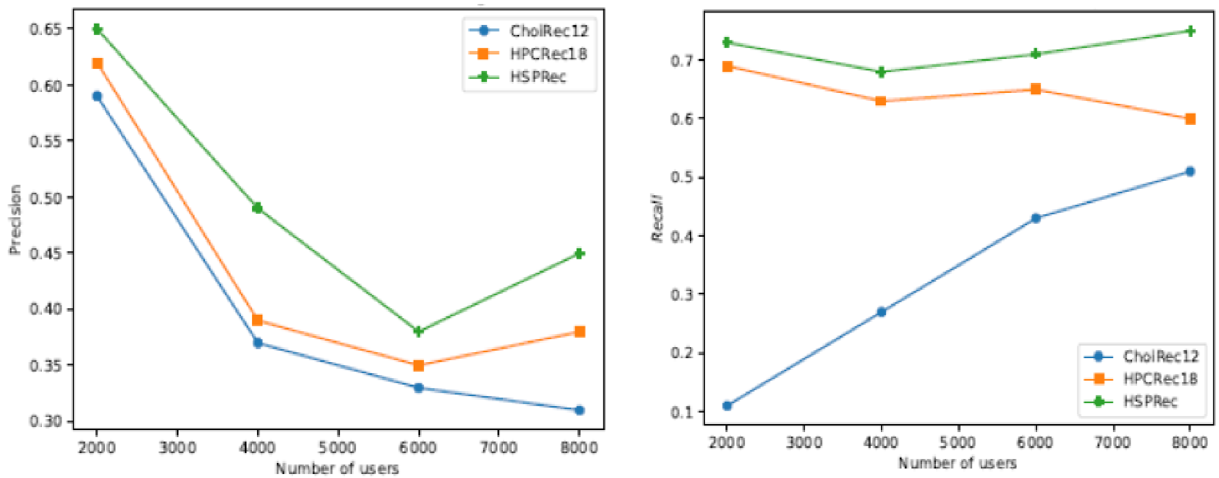
Here (Fig 4.1), the performance of SP-based E-commerce RS like ChoiRec12, HPCRec18, and HSPRec19 recommendation systems against traditional CF algorithm was evaluated in terms of quality of ratings prediction with respect to predictive accuracy measure MAE metric by varying number of users (left side graph) and nearest neighbors (right side graph). MAE compares the predicted ratings to actual user ratings over a test sample in a recommendation system and is defined as the average absolute difference between predicted ratings and actual ratings.

Figure 4.1. Evaluation of Quality of the ratings prediction (Bhatta, 2019)



Here (Fig 4.2), the performance of SP-based E-commerce RS like ChoiRec12, HPCRec18, and HSPRec19 systems were evaluated in terms of quality of recommendations generated by varying number of users with respect to classification accuracy measures such as precision and recall, which evaluates the frequency of the system making correct/incorrect decisions. Precision is the fraction of all recommended items that are relevant, and Recall is the fraction of all relevant items that were recommended.

Figure 4.2. Evaluation of Quality of the recommendations (Bhatta, 2019)



The results obtained from the experimental comparative analysis of Traditional CF, ChoiRec12, HPCRec18 & HSPRec19 systems conducted by (Bhatta, Ezeife & Butt, 2019) have shown that HSPRec19 system performed the best in comparison to the other recommendation systems as it used SPM (GSP algorithm) to discover frequent historical sequential patterns and analyzed the clickstream behavior for improving the consequential bond between clicks and purchases to enhance user-item frequency matrix quantitatively and qualitatively to generate a rich user-item matrix for CF thereby, resulting better recommendations in terms of reduced data sparsity and improved recommendation accuracy, scalability, diversity and novelty. Thus, out of all the reviewed SP-based E-commerce RS, I would suggest using HSPRec19 system for the purpose of recommendation in a real-life application scenario. Out of the evaluation metrics MAE, Precision and Recall, the most important metric for comparing recommendation systems is MAE because of its ability to measure the average absolute deviation (error) between the system’s predicted rating and the actual rating assigned by the user.

CHAPTER 5: CONCLUSIONS & FUTURE WORK

Recommendation Systems open new opportunities of retrieving personalized information on the internet by enabling the users to have access to products and services which are not readily available to users on the system. Many recommendation systems neglect sequential patterns during recommendation. Thus, to verify the necessity of sequential patterns in E-commerce recommendation systems, a survey of the existing SP-based E-commerce RS is conducted, and a taxonomy is developed that classifies these applications by their input, output, recommendation method and performance factors like reducing data sparsity, improving scalability of recommendation systems and improving accuracy & novelty of recommendations. Furthermore, after performing a comparative analysis of traditional CF against few of the surveyed SP-based E-commerce RS, the results have proved that the hybridization of SPM with CF by integrating sequential patterns into the user-item rating matrix input, improved the recommendation quality in terms of accuracy, diversity and novelty. Additionally, we would like to direct the reader to open research subjects that warrant future works in the area of SP-based E-commerce RS and the ideas for future work in this direction include:

1. None of the reviewed studies exactly measured the level of probability of purchase determined by each SP, instead the general mid-way of 50% (Bhatta, Ezeife & Butt, 2019) was used for example. Hence, more information (such as the frequency of the patterns occurring together) in the historical data should be used to determine the exact level of probability of purchase (e.g., 0.5 to 1.0) for each SP.
2. More possible ways of incorporating click stream sequences/patterns into the User-Item rating matrix should be found with the use of consequential bond to improve the input User-Item rating quality. Also, additional information such as contextual data (e.g., time of the year, such as season or month, or day of the week etc.) should be integrated into user-item preferences.
3. Incorporating the factor of profit or utility for finding patterns (apart from just finding the frequent sequential patterns) from historical purchase data results in profitable recommendations. Thus, high utility sequential patterns should be integrated into the recommendation generation processes.
4. In the real world, items purchased by a user during a certain time period are often from multi-domains rather than one domain. Essentially, there are some sequential dependencies between items from different domains (e.g., the purchase of a car insurance after the

purchase of a car). Such cross-domain sequential dependencies are ignored in most sequential pattern-based recommendation systems. Therefore, cross-domain recommendation systems are another promising research direction to generate more accurate recommendations by leveraging information and diverse recommendations from different domains.

5. Apart from the available multiple actions related to certain items in the e-commerce domain (e.g., add-to wishlist, add-to-cart), there are also other relevant user actions like search or category navigation which are not considered to a large extent in today's research. Thus, this richer type of information should also be incorporated in the recommendation generation process.
6. There is a need for extensive research for extending the capabilities of existing approaches by integrating the factors such as user preference drift, items popularity drift, change of product popularities, dynamic interest within the community, seasonal effects, changes in rating scales and detecting long-term and transient or short-term behavioral patterns.
7. Incorporating multi-criteria information like user demographics and gender can have a dramatic effect on the interpretation and utility of recommendation results as it allows the users to express more differentiated opinions by allowing separate ratings for different aspects or dimensions of an item.
8. Most shopping behaviors in the real-world are continuous rather than isolated events. In other words, there are sequential interactions between a user and the shopping platform (e.g., Amazon). However, the existing SRSs often neglect such interactions and only generate recommendations for one action at a single time step. Thus, generating multi-time step recommendations by incorporating user-seller interactions is a promising research direction.
9. Another good line of future research is the evaluation strategy used to assess the performance of sequential pattern-based recommendation systems, as all the reviewed studies were evaluated based on the offline approaches. Although the offline evaluation is of lower cost with no bias of response from active user involvements as in the case of online and user studies, the results mostly contradict when applied in real-life applications with the online and user studies evaluations. Therefore, there is a huge need for more research on the evaluation strategies to compare performance based on different performance measures other than accuracy and offline evaluation, like real-time, novelty, coverage, serendipity and diversity etc.

REFERENCES

Adomavicius, G., & Tuzhilin, A. (2005). Towards the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17, 6, , 734–749.

Aggarwal, C. C. (2016). An introduction to recommender systems. *In Recommender Systems, Springer*, 1-28.

Agrawal, R., & Srikant, R. (1994). Fast Algorithms for Mining Association Rules in Large Databases. *In Proceedings of the 20th International Conference on Very Large Data Bases (VLDB '94)*, 487-499.

Agrawal, R., & Srikant, R. (1995). Mining Generalized Association Rules. *In Proceedings of the 21st International Conference on Very Large Data Bases (VLDB '95)*, 407-419.

Agrawal, R., & Srikant, R. (1996). Mining quantitative association rules in large relational tables. *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data - SIGMOD '96*, 1-12.

Ayres, J., Flannick, J., Gehrke, J., & Yiu, T. (2002). Sequential pattern mining using a bitmap representation. *In Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 429–435.

Balabanović, M., & Shoham, Y. (1997). Fab: Content-based, collaborative recommendation. *Communications of the ACM*, 40(3), 66-72.

Batmaz, Z., Yurekli, A., Bilge, A., & Kaleli, C. (2018). A review on deep learning for recommender systems: Challenges and remedies. *Artificial Intelligence Review*, 52(1), 1-37.

Bhatta, R., Ezeife, C., & Butt, M. N. (2019). Mining sequential patterns of historical purchases for e-commerce recommendation. *In International conference on big data analytics and knowledge discovery*, 57-72.

Bhatta, R. (2019). "*Discovering E-commerce Sequential Data Sets and Sequential Patterns for Recommendation*" (Master's Thesis, University of Windsor, Windsor, Canada). Retrieved from https://scholar.uwindsor.ca/etd/7686.

Bokde, D., Girase, S., & Mukhopadhyay, D. (2015). Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey. *Procedia Computer Science*, *49*, 136-146.

Bridge, D., Goker, M., McGinty, L., & Smith, B. (2005). Case-based recommender systems. Knowl. *Eng. Rev*., 20(3), 315-320.

Bucklin, R. E., & Sismeiro, C. (2003). A model of web site browsing behavior estimated on clickstream data. *Journal of marketing research*, 40, 249-267.

Burke R., 2002. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*. 12(4), 331-370.

Chen, M. S., Han, J., & Yu, P. S. (1996). Data mining: an overview from a database perspective. *IEEE Transactions on Knowledge and data Engineering*, 8, 866-883.

Chen, S., Moore, J. L., Turnbull, D., & Joachims, T. (2012). Playlist prediction via metric embedding. *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* - KDD '12, 714-722.

Chen, Y. L., Kuo, M. H., Wu, S. Y., & Tang, K. (2009). Discovering recency, frequency, and monetary (RFM) sequential patterns from customer's purchasing data. *Electronic Commerce Research and Applications*, 8(5), 241-251.

Cheng, C., Yang, H., Lyu, M., & King, I. (2013). Where you like to go next: Successive point-of-interest recommendation. *In Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence* - IJCAI '13, 2605-2611.

Chesnais, P. R., Mucklo, M. J., & Sheena, J. A. (1995). The Fishwrap personalized news system. *In Proceedings of the Second International Workshop on Community Networking Integrated Multimedia Services to the Home* (pp. 275-282). IEEE.

Chiu, D.-Y., Wu, Y.-H., & Chen, A. L. P. (2004). An efficient algorithm for mining frequent sequences by a new strategy without support counting. *In Proceedings of the 20th International Conference on Data Engineering*. 375–386.

Cho, Y. B., Cho, Y. H., & Kim, S. H. (2005). Mining changes in customer buying behavior for collaborative recommendations. *Expert Systems with Applications*, 28 (2), 359-369.

Choi, K., Yoo, D., Kim, G., & Suh, Y. (2012). A hybrid online-product recommendation system: Combining implicit rating-based collaborative filtering and sequential pattern analysis. *Electronic Commerce Research and Applications*, 11 (4), 309-317.

Chun, J., Oh, J. Y., Kwon, S., & Kim, D. (2004). Simulating the Effectiveness of Using Association Rules for Recommendation Systems. *Lecture Notes in Computer Science Systems Modeling and Simulation: Theory and Applications*, 306-314.

Covington, P., Adams, J., & Sargin, E. (2016). Deep Neural Networks for YouTube Recommendations. *In Proceedings of the 10th ACM Conference on Recommender Systems RecSys* '16, 191-198.

El-Sayed, M., Ruiz, C., & Rundensteiner, E. A. (2004). FS-Miner: Efficient and incremental mining of frequent sequence patterns in web logs. *In Proceedings of the 6th Annual ACM International Workshop on Web Information and Data Management. ACM, New York*, 128–135.

Ezeife, C. I., & Lu, Y. (2005). PLWAP sequential mining: Open source code. *In Proceedings of the 1st International Workshop on Open Source Data Mining: Frequent Pattern Mining Implementation (SIGKDD), ACM*, New York, 26–35.

Fang, Z., Zhang, L., & Chen, K. (2016). A behavior mining based hybrid recommender system. *2016 IEEE International Conference on Big Data Analysis (ICBDA)*.

Felfernig, A., & Burke, R. (2008). Constraint-based recommender systems. *Proceedings of the 10th International Conference on Electronic Commerce - ICEC* '08, 3, 1-10.

Felfernig, A., Friedrich, G., Jannach, D., & Zanker, M. (2011). Developing Constraint-based Recommenders. *Recommender Systems Handbook*, 187-215.

Goldberg, K., Roeder, T., Gupta, D., & Perkins, C. (2001) Eigentaste: a constant time collaborative filtering algorithm. *Information Retrieval*, 4(2), 133–151.

Han, J., Kamber, M., & Pei, J. (2011). Data mining: Concepts and techniques. *Amsterdam: Elsevier*.

Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U., & HSU, M.-C. (2000). Freespan: Frequent pattern projected sequential pattern mining. *In Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, New York*, 355–359.

Huang, C. L., & Huang, W. L. (2009). Handling sequential pattern decay: Developing a two-stage collaborative recommender system. *Electronic Commerce Research and Applications*, 8(3), 117–129.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. ACM computing surveys (CSUR), 31, 264-323.

Jamali, S., & Navaei, Y. D. (2016). A two-level Product Recommender for E-commerce Sites by Using Sequential Pattern Analysis. *International Journal of Integrated Engineering*, 8(1).

Jannach, D., &. Adomavicius, G. (2016). Recommendations with a Purpose. *Proceedings of the 10th ACM Conference on Recommender Systems*, 7-10.

Jannach, D., & Jugovac, M. (2019). Measuring the Business Value of Recommender Systems. *ACM Transactions on Management Information Systems*, 10(4), 1-23.

Jannach, D., Lerche, L., & Jugovac, M. (2015). Adaptation and Evaluation of Recommendations for Short-term Shopping Goals. *Proceedings of the 9th ACM Conference on Recommender Systems - RecSys* '15, 211-218.

Jindal, H., & Sardana, N. (2020). Web navigation prediction based on dynamic threshold heuristics. *Journal of King Saud University - Computer and Information Sciences*, 1-11.

Joaquin, D., and Naohiro, I. Memory-based weighted-majority prediction. *In ACM SIGIR' 99 Workshop on Recommender Systems: Algorithms and, Evaluation*, 1999.

Kaymak, U. (2001). Fuzzy target selection using RFM variables. *Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference* (Cat. No. 01TH8569), 2, 1038-1043.

Kazienko, P., & Pilarczyk, M. (2008). Hyperlink Recommendation Based on Positive and Negative Association Rules. *New Generation Computing*, 26(3), 227-244.

Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., & Riedl, J. (1997). GroupLens: Applying collaborative filtering to Usenet news. *Communications of the ACM*, 40(3), 77-87.

Lang, K. (1995). NewsWeeder: Learning to Filter Netnews. *Machine Learning Proceedings* 1995, 331-339.

Lim, K. H., Chan, J., Karunasekera, S., & Leckie, C. (2018). Tour recommendation and trip planning using location-based social media: A survey. *Knowledge and Information Systems*, 60(3), 1247-1275.

Li, C., Niu, X., Luo, X., Chen, Z., & Quan, C. (2019). A Review-Driven Neural Model for Sequential Recommendation. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 2886-2872.

Li, J., Wang, Y., & Mcauley, J. (2020). Time Interval Aware Self-Attention for Sequential Recommendation. *Proceedings of the 13th International Conference on Web Search and Data Mining*, 322-330.

Liu, D., Lai, C., & Lee, W. (2009). A hybrid of sequential rules and collaborative filtering for product recommendation. *Information Sciences*, 179(20), 3505-3519.

Liu, J., Dolan, P., & Pedersen, E. R. (2010). Personalized news recommendation based on click behavior. *Proceedings of the 15th International Conference on Intelligent User Interfaces* - IUI '10, 31-40.

Liu, Y., Liao, W.-k., & Choudhary, A. (2005). A two-phase algorithm for fast discovery of high utility itemsets. *In Pacific-asia conference on knowledge discovery and data mining* (pp. 689-695).

Mabroukeh, N. R. & Ezeife, C. I. (2010). A taxonomy of sequential pattern mining algorithms. *ACM Comput. Surv*. 43, 1, 3.

Masseglia, F., Poncelet, O., & Cicchetti, R. (1999). An efficient algorithm for web usage mining. *Network Inform. Syst*. J. 2, 571–603.

Mooney, C. H., & Roddick, J. F. (2013). Sequential pattern mining -- approaches and algorithms. *ACM Computing Surveys*, 45(2), 1-39.

Pazzani, M. J., & Billsus, D. (2007). Content-Based Recommendation Systems. *Lecture Notes in Computer Science*, 325–341.

Pauws, S., Verhaegh, W., & Vossen, M. (2006). Music playlist generation by adapted simulated annealing. *Information Sciences*, 178(3), 647-662.

Pei, J., Han, J., Mortazavi-Asl, B., & Pinto, H. (2001). PrefixSpan: Mining sequential patterns efficiently by prefix-projected pattern growth. *In Proceedings of the International Conference on Data Engineering*. 215–224.

Pei, J., Han, J., Mortazavi-Asl, B., & Zhu, H. (2000). Mining access patterns efficiently from web logs. *In Knowledge Discovery and Data Mining. Current Issues and New Applications. Lecture Notes Computer Science*, *vol*. 1805, *Springer, Berlin*, 396–407.

Quadrana, M., Cremonesi, P., &, Jannach, D. (2018). Sequence-aware Recommender Systems. *ACM Computing Surveys*, 1(1), 1-35.

Rendle, S., Freudenthaler, C., & Schmidt-Thieme, L. (2010). Factorizing personalized Markov chains for next-basket recommendation. *Proceedings of the 19th International Conference on World Wide Web - WWW* '10, 811-820.

Ricci, F., Rokach, L., Shapira, B., & Kantor, P. B. (Eds.). (2011). Recommender Systems Handbook.

Ruocco, M., Skrede, O. S., & Langseth, H. (2017). Inter-Session Modeling for Session-Based Recommendation. *Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems* - DLRS 2017, 24-31.

Saini, S., Saumya, S., & Singh, J. P. (2017). Sequential purchase recommendation system for e-commerce sites. *In IFIP International Conference on Computer Information Systems and Industrial Management* (pp. 366-375).

Salakhutdinov, R., Mnih, A., & Hinton, G. (2007). Restricted Boltzmann machines for collaborative filtering. *Proceedings of the 24th International Conference on Machine Learning - ICML '07*, 791-798.

Salter, J., & Antonopoulos, N. (2006). CinemaScreen Recommender Agent: *Combining Collaborative and Content-Based Filtering. IEEE Intelligent Systems, 21(1)*, 35-41.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). Analysis of recommendation algorithms for e-commerce. *Proceedings of the 2nd ACM Conference on Electronic Commerce - EC* '00, 158-167.

Schafer, J. B., Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative filtering recommender systems. *In The adaptive web* (pp. 291-324). *Springer, Berlin, Heidelberg*.

Shani, G., Heckerman, D., & Brafman, R. (2002). An MDP-based recommender system. *In Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence - UAI*'02, 453-460.

Shin, D., Cetintas, S., Lee, K., & Dhillon, I. S. (2015). Tumblr Blog Recommendation with Boosted Inductive Matrix Completion. *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management - CIKM* '15, 203-212.

Shu, J., Shen, X., Liu, H., Yi, B., & Zhang, Z. (2017). A content-based recommendation algorithm for learning resources. *Multimedia Systems*, 24(2), 163-173.

Si, L., & Jin, R. (2003). Flexible mixture model for collaborative filtering. *In Proceedings of the Twentieth International Conference on International Conference on Machine Learning - ICML*'03, 704-711.

Singhal, A., Sinha, P., & Pant, R. (2017). Use of Deep Learning in Modern Recommendation System: A Summary of Recent Works. *International Journal of Computer Applications,* 180(7), 17-22.

Song, S., Hu, H., & Jin, S. (2005). HVSM: A new sequential pattern mining algorithm using bitmap representation. *In Advanced Data Mining and Applications. Lecture Notes in Computer Science*, *vol*. 3584, *Springer, Berlin*, 455–463.

Su, X., & Khoshgoftaar, T. M. (2009). A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*, 2009, 1-19.

Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., & Jiang, P. (2019). BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 1441-1450.

Wang, S., Hu, L., Wang, Y., Cao, L., Sheng, Q. Z., & Orgun, M. (2019). Sequential Recommender Systems: Challenges, Progress and Prospects. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*.

Wang, H., Shi, X., & Yeung, D. (2015). Collaborative recurrent autoencoder: Recommend while learning to fill in the blanks. *In Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS*'16), 415-423.

Wei, K., Huang, J., & Fu, S. (2007). A survey of e-commerce recommender systems. *Service systems and service management, 2007 international conference on,* (pp.1-5).

Wei Song & Kai Yang. (2014). Personalized Recommendation Based on Weighted Sequence Similarity. *In Practical Applications of Intelligent Systems*. *Springer*, (657–666).

Xiao, Y., & Ezeife, C. I. (2018). E-Commerce Product Recommendation Using Historical Purchases and Clickstream Data. *Big Data Analytics and Knowledge Discovery Lecture Notes in Computer Science*, (pp. 70–82).

Xiao, Y. (2018). "*Recommending Best Products from E-commerce Purchase History and User Click Behavior Data*" (Master's Thesis, University of Windsor, Windsor, Canada). Retrieved from https://scholar.uwindsor.ca/etd/7455.

Xiao, T., Liang, S., & Meng, Z. (2019). Hierarchical Neural Variational Model for Personalized Sequential Recommendation. *The World Wide Web Conference on* - *WWW* '19, 3377-3383.

Xu, M., Liu, F., & Xu, W. (2019). A Survey on Sequential Recommendation. *In 2019 6th International Conference on Information Science and Control Engineering (ICISCE)* (pp. 106-111). IEEE.

Xu, Z., Lukasiewicz, T., Chen, C., Miao, Y., & Meng, X. (2017) Tag-aware personalized recommendation using a hybrid deep model. *In: Proceedings of the 26th international joint conference on artificial intelligence, Melbourne, Australia*, pp 3196–3202.

Yang, Z., Wang, Y., & Kitsuregawa, M. (2007). LAPIN: Effective sequential pattern mining algorithms by last position induction for dense databases. *In Advances in Databases: Concepts, Systems and Applications. Lecture Notes in Computer Science, vol*. 4443, 1020–1023.

Yao, H., & Hamilton, H. J. (2006). Mining itemset utilities from transaction databases. *Data & Knowledge Engineering*, 59(3), 603–626.

Yap, G., Li, X., & Yu, P. S. (2012). Effective Next-Items Recommendation via Personalized Sequential Pattern Mining. *Database Systems for Advanced Applications Lecture Notes in Computer Science*, 48-64.

Yates, B.R., Jiang, D., Silvestri, F., & Harrison, B. (2015). Predicting the Next App That You Are Going To Use. *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining - WSDM* '15, 285-294.

Zaki, M. J. (2001). SPADE: An efficient algorithm for mining frequent sequences. *Mach. Learn*. 42, 31–60.

Zhang, F., Yuan, N., Lian, D., & Xie, X. (2016) Collaborative knowledge base embedding for recommender systems. *In: Proceedings of the 22nd ACM SIGKDD International conference on knowledge discovery and data mining, ACM, San Francisco, CA, USA*, pp 353–362

Zhang, Y., Dai, H., Xu, C., Feng, J., Wang, T., Bian, J., ... & Liu, T. Y. (2014). Sequential click prediction for sponsored search with recurrent neural networks. *In Twenty-Eighth AAAI Conference on Artificial Intelligence*.

Zhao, Q., & Bhowmick, S. S. (2003). Association rule mining: A survey. Nanyang Technological University, Singapore, 135.

Zhao Y, Wang J, Wang F (2015) Word embedding based retrieval model for similar cases recommendation. *In: Proceedings of 2015 Chinese automation congress, Wuhan, China*, pp 2268–2272.

Zhou, B., Hui, S., & Chang, K. (2004). An intelligent recommender system using sequential Web access patterns. *IEEE Conference on Cybernetics and Intelligent Systems*.

Zhou J, Albatal R, Gurrin C (2016) Applying visual user interest profiles for recommendation and personalisation. *In: Proceedings of the 22nd international conference on multimedia modeling, Miami, FL, USA*, pp 361–366

Zuo, Y., Gong, M., Zeng, J., Ma, L., &, Jiao, L. (2015). Personalized Recommendation Based on Evolutionary Multi-Objective Optimization [Research Frontier]. *IEEE Computational Intelligence Magazine*, 10(1), 52-62.

VITA AUCTORIS

NAME	Hemni Sri Rajeswari Karlapalepu
PLACE OF BIRTH	Andhra Pradesh, India
YEAR OF BIRTH	1995
EDUCATION	Bhashyam High School, Guntur, AP, India (2010 - 2012)
	Gayatri Vidya Parishad College of Engineering for Women's, B.Tech., AP, India (2012 - 2016)
	University of Windsor, M.Sc., Windsor, ON, Canada (2019 - 2020)